ifm electronic

CE

ecomat mobile

Systemhandbuch
BasicController

ecomat100

CR0401
CR0403

CoDeSys® V2.3

Target V02

Deutsch

© ifm electronic gmbh | www.ifm.com

7390709_06_DE 2012-09-05

Inhalt

1 Über diese Anleitung 5
1.1 Was bedeuten die Symbole und Formatierungen? 5
1.2 Wie ist diese Dokumentation aufgebaut? 6
1.3 Historie der Anleitung (BC) 7

2 Sicherheitshinweise 8
2.1 Wichtig! 8
2.2 Welche Vorkenntnisse sind notwendig? 9

3 Systembeschreibung 10
3.1 Angaben zum Gerät 10
3.1.1 Unterschiede in der Gerätefamilie 10
3.1.2 Zubehör 13
3.2 Angaben zur Software 14
3.3 Steuerungskonfiguration 15

4 Konfigurationen 16
4.1 Programmiersystem einrichten 16
4.1.1 Programmiersystem manuell einrichten 16
4.1.2 Programmiersystem über ifm-Templates einrichten 21
4.2 Funktionskonfiguration der Ein- und Ausgänge 22
4.2.1 Eingänge konfigurieren 23
4.2.2 Ausgänge konfigurieren 29
4.3 Hinweise zur Anschlussbelegung 33

5 Betriebszustände und Betriebssystem 35
5.1 Betriebszustände 35
5.1.1 INIT-Zustand (Reset) 35
5.1.2 RUN-Zustand 36
5.1.3 STOP-Zustand 36
5.1.4 ERROR-Zustand 37
5.1.5 FATAL-ERROR-Zustand 37
5.2 Status-LED 38
5.3 Betriebssystem 39
5.3.1 Betriebssystem laden 39

6 Begrenzungen und Programmierhinweise 41
6.1 Leistungsgrenzen des Geräts 41
6.1.1 CPU-Frequenzen 41
6.1.2 Verhalten des Watchdog 42
6.1.3 Verfügbarer Speicher 43
6.1.4 Verfügbarer Speicher 43
6.2 Programmierhinweise für CoDeSys-Projekte 44
6.2.1 FB, FUN, PRG in CoDeSys 44
6.2.2 Zykluszeit beachten! 45
6.2.3 Speichern 45

© ifm electronic gmbh | www.ifm.com

6.2.4 Applikations-Programm erstellen ... 45
6.2.5 ifm-Maintenance-Tool nutzen ... 47
6.2.6 Verteilen der Applikations-Software ... 47

7 Fehlercodes und Diagnoseinformationen 48

7.1 Übersicht ... 48
7.2 Reaktion auf System-Fehler ... 48
7.2.1 Beispielablauf für Reaktion auf System-Fehler ... 48

8 CAN einsetzen 49

8.1 Allgemeines zu CAN ... 49
8.1.1 Topologie ... 49
8.1.2 CAN-Schnittstellen ... 50
8.1.3 Verfügbare CAN-Schnittstellen und CAN-Protokolle ... 50
8.1.4 System-Konfiguration ... 52
8.2 Physikalische Anbindung des CAN ... 53
8.2.1 Netzaufbau ... 53
8.2.2 CAN-Buspegel ... 54
8.2.3 Busleitungslänge ... 55
8.2.4 Leitungsquerschnitte ... 56
8.3 CAN-Datenaustausch ... 57
8.3.1 Hinweise ... 58
8.3.2 CAN-Funktionen im IEC-User-Zyklus ... 60
8.4 CAN-Bausteine auf Schicht 2 (RAW-CAN) ... 64
8.4.1 Hinweise ... 64
8.4.2 FBs für RAW-CAN (Verwaltung) ... 66
8.4.3 FBs für RAW-CAN (Nachrichten senden) ... 73
8.4.4 FBs für RAW-CAN (Nachrichten empfangen) ... 80
8.4.5 FBs für RAW-CAN (Remote-Zugriff) ... 91
8.5 CAN-Bausteine nach SAE J1939 ... 96
8.5.1 CAN für die Antriebstechnik ... 97
8.5.2 Bausteine für J1939 (Verwaltung) ... 102
8.5.3 Bausteine für J1939 (Senden) ... 111
8.5.4 Bausteine für J1939 (Empfangen) ... 121
8.6 CAN-Bausteine nach CANopen ... 130
8.6.1 Technisches zu CANopen ... 130
8.6.2 ifm-CANopen-SDO-Bausteine ... 164
8.6.3 CANopen-Netzwerk-Management ... 175
8.7 CAN-Fehler und Fehlerbehandlung ... 199
8.7.1 Aufbau einer EMCY-Nachricht ... 199
8.7.2 Übersicht CANopen Error-Codes ... 201
8.7.3 CANopen-Bausteine zur Fehlerbehandlung ... 204

9 Ein-/Ausgangs-Funktionen 211

9.1 Eingangswerte verarbeiten ... 211
9.1.1 INPUT ... 212
9.2 Zählerfunktionen zur Frequenz- und Periodendauermessung ... 215
9.2.1 FASTCOUNT ... 216
9.2.2 INC_ENCODER ... 218
9.2.3 PERIOD ... 221
9.3 Ausgangsfunktionen allgemein ... 223
9.3.1 OUTPUT ... 224
9.4 PWM-Funktionen ... 227
9.4.1 Verfügbarkeit von PWM ... 227
9.4.2 PWM-Signalverarbeitung ... 228
9.4.3 Stromregelung mit PWM ... 232

© ifm electronic gmbh | www.ifm.com

10 Daten verwalten 235

10.1 Systemzeit lesen / schreiben ... 235
10.1.1 TIMER_READ_US ... 236
10.2 Daten im Speicher sichern, lesen und wandeln ... 237
10.2.1 Automatische Datensicherung ... 237
10.2.2 Manuelle Datensicherung ... 238
10.3 Datenzugriff und Datenprüfung ... 244
10.3.1 GET_IDENTITY ... 245
10.3.2 SET_IDENTITY ... 246
10.3.3 SET_PASSWORD ... 247
10.4 Dateien verwalten ... 248
10.4.1 Dateifunktionen ... 248

11 SPS-Zyklus optimieren 253

11.1 Zykluszeit steuern ... 253
11.1.1 PLCPRG_DELAY ... 254

12 LED, Buzzer, Visualisierung 256

12.1 LED ansteuern ... 256
12.1.1 SET_LED ... 257

13 Anhang 259

13.1 Adressbelegung und E/A-Betriebsarten ... 259
13.1.1 Adressbelegung Ein-/Ausgänge ... 260
13.1.2 Mögliche Betriebsarten Ein-/Ausgänge (CR0401) ... 262
13.1.3 Mögliche Betriebsarten Ein-/Ausgänge (CR0403) ... 263
13.2 Systemmerker ... 264
13.3 CANopen-Tabellen ... 265
13.3.1 IDs (Adressen) in CANopen ... 265
13.3.2 Aufbau von CANopen-Meldungen ... 266
13.3.3 Bootup-Nachricht ... 271
13.3.4 Netzwerk-Management (NMT) ... 272
13.3.5 CANopen Error-Code ... 276
13.4 Arbeiten mit dem User-Flash-Speicher ... 279
13.4.1 Flash-Speicher – was ist das? ... 279
13.4.2 CSV-Datei – was ist das? ... 280
13.4.3 CSV-Datei erzeugen und übertragen ... 280
13.5 Übersicht der verwendeten Dateien und Bibliotheken ... 294
13.5.1 Dateien und Bibliotheken im Gerät installieren ... 294
13.5.2 Allgemeine Übersicht ... 295
13.5.3 Wozu dienen die einzelnen Dateien und Bibliotheken? ... 297

14 Begriffe und Abkürzungen 301

15 Index 318

16 ifm weltweit • ifm worldwide • ifm à l'échelle internationale 325

© ifm electronic gmbh | www.ifm.com

1 Über diese Anleitung

Inhalt

Was bedeuten die Symbole und Formatierungen?.....5
Wie ist diese Dokumentation aufgebaut?.....6
Historie der Anleitung (BC).....7

202

Im ergänzenden "Programmierhandbuch CoDeSys V2.3" erhalten Sie weitergehende Informationen über die Nutzung des Programmiersystems "CoDeSys for Automation Alliance".
Dieses Handbuch steht auf der **ifm**-Homepage als kostenloser Download zur Verfügung:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm
Handbücher und Online-Hilfen für **ecomat*mobile*** finden Sie auch hier:
→ **ecomat*mobile***-DVD "Software, tools and documentation"

Niemand ist vollkommen. Wenn Sie uns Verbesserungsvorschläge zu dieser Anleitung melden, erhalten Sie von uns ein kleines Geschenk als Dankeschön.

© Alle Rechte bei **ifm electronic gmbh**. Vervielfältigung und Verwertung dieser Anleitung, auch auszugsweise, nur mit Zustimmung der **ifm electronic gmbh**.

Alle auf unseren Seiten verwendeten Produktnamen, -Bilder, Unternehmen oder sonstige Marken sind Eigentum der jeweiligen Rechteinhaber:
- AS-i ist Eigentum der AS-International Association, (→ www.as-interface.net)
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ www.can-cia.org)
- CoDeSys™ ist Eigentum der 3S – Smart Software Solutions GmbH, Deutschland (→ www.3s-software.com)
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- IO-Link® (→ www.io-link.com) ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ www.microsoft.com)
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ www.profibus.com)
- PROFINET® ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® ist Eigentum der →Microsoft Corporation, USA

1.1 Was bedeuten die Symbole und Formatierungen?

203

Folgende Symbole oder Piktogramme verdeutlichen Ihnen unsere Hinweise in unseren Anleitungen:

⚠ WARNUNG

Tod oder schwere irreversible Verletzungen sind möglich.

⚠ VORSICHT

Leichte reversible Verletzungen sind möglich.

ACHTUNG

Sachschaden ist zu erwarten oder möglich.

	Wichtige Hinweise auf Fehlfunktionen oder Störungen
	Weitere Hinweise
► ...	Handlungsaufforderung

> ...	Reaktion, Ergebnis
→ ...	"siehe"
abc	Querverweis
[...]	Bezeichnung von Tasten, Schaltflächen oder Anzeigen

1.2 Wie ist diese Dokumentation aufgebaut?

204

Diese Dokumentation ist eine Kombination aus verschiedenen Anleitungstypen. Sie ist eine Lernanleitung für den Einsteiger, aber gleichzeitig auch eine Nachschlageanleitung für den versierten Anwender.

Und so finden Sie sich zurecht:

- Um gezielt zu einem bestimmten Thema zu gelangen, benutzen Sie bitte das Inhaltsverzeichnis.
- Mit dem Stichwortregister "Index" gelangen Sie ebenfalls schnell zu einem gesuchten Begriff.
- Am Anfang eines Kapitels geben wir Ihnen eine kurze Übersicht über dessen Inhalt.
- Abkürzungen und Fachbegriffe → Anhang.

Bei Fehlfunktionen oder Unklarheiten setzen Sie sich bitte mit dem Hersteller in Verbindung:
→ www.ifm.com > Land wählen > [Kontakt].

Wir wollen immer besser werden! Jeder eigenständige Abschnitt enthält in der rechten oberen Ecke eine Identifikationsnummer. Wenn Sie uns über Unstimmigkeiten unterrichten wollen, dann nennen Sie uns bitte diese Nummer zusammen mit Titel und Sprache dieser Dokumentation. Vielen Dank für Ihre Unterstützung!

Im Übrigen behalten wir uns Änderungen vor, so dass sich Abweichungen vom Inhalt der vorliegenden Dokumentation ergeben können. Die aktuelle Version finden Sie auf der ifm-Homepage:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm
⇨ Unsere Online-Hilfen sind meist "tagesaktuell".
⇨ Die PDF-Handbücher aktualisieren wir nur in großen zeitlichen Abständen.

© ifm electronic gmbh | www.ifm.com

1.3 Historie der Anleitung (BC)

9193

Was hat sich wann in dieser Anleitung geändert? Ein Überblick:

Datum	Thema	Änderung
2011-01-10	OUTPUT (FB)	Parameter für Eingang MODE korrigiert
2011-02-07	OUTPUT (FB)	Parameter für Eingang MODE korrigiert
2011-02-14	TIMER_READ_US (FB)	Umrechnung max. Zählwert korrigiert
2011-02-16	Adressbelegung der Ausgänge für CR0401	ab V01.02.zz anders als bei CR0403
2011-03-16	PWM-Ausgänge	PWM-Ausgänge nicht parallel betreiben!
2011-04-13	CANopen Übersicht	neu: CANopen-Tabellen im Anhang
2011-05-24	CANopen-Slave einfügen und konfigurieren	Screenshot korrigiert
2011-07-14	Verhalten des Watchdog	Korrektur und Erweiterung
2011-07-14	diverse Funktionsbausteine	- zulässige Parameterwerte ergänzt - FiFo-Werte korrigiert
2011-07-14	Ein- und Ausgänge	- IEC-Adressen korrigiert - Konfiguration der Modi - Verhalten der Ausgänge im Fehlerfall
2011-09-01	Parameter in FB J1939_DM1TX	- Parameter SPN_CM - Parameter NUMBER
2011-09-17	diverse	- FB INPUT: Filterwerte - FB OUTPUT: Filterwerte - Spannungsversorgung bei Programm-Upload - CAN FiFo-Größe - MODE ändern im laufenden Betrieb
2011-10-07	Kapitel "Fehlercodes und Diagnoseinformationen"	System liefert keine Fehler-Systemmerker
2012-07-16	CURRENT_CONTROL (FB)	Hinweis zu den Parameter-Werten
2012-07-16	FLASHWRITE (FB)	nicht in diesem Gerät
2012-07-24	Laufzeitsystem	Upgrade auf V02

© ifm electronic gmbh | www.ifm.com

2 Sicherheitshinweise

Inhalt

Wichtig! 8
Welche Vorkenntnisse sind notwendig? 9

213

2.1 Wichtig!

6091

11212

Mit den in dieser Anleitung gegebenen Informationen, Hinweisen und Beispielen werden keine Eigenschaften zugesichert. Die abgebildeten Zeichnungen, Darstellungen und Beispiele enthalten weder Systemverantwortung noch applikationsspezifische Besonderheiten.

► Die Sicherheit der Maschine/Anlage muss auf jeden Fall eigenverantwortlich durch den Hersteller der Maschine/Anlage gewährleistet werden.

► Beachten Sie die nationalen Vorschriften des Landes, in welchem die Maschine/Anlage in Verkehr gebracht werden soll!

⚠ WARNUNG

Sach- oder Körperschäden sind möglich bei Nichtbeachten der Hinweise in dieser Anleitung!
Die **ifm electronic gmbh** übernimmt hierfür keine Haftung.

► Die handelnde Person muss vor allen Arbeiten an und mit diesem Gerät die Sicherheitshinweise und die betreffenden Kapitel dieser Anleitung gelesen und verstanden haben.

► Die handelnde Person muss zu Arbeiten an der Maschine/Anlage autorisiert sein.

► Beachten Sie die Technischen Daten der betroffenen Geräte!
Das aktuelle Datenblatt finden Sie auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Technische Daten im PDF-Format]

► Beachten Sie die Montage- und Anschlussbedingungen sowie die bestimmungsgemäße Verwendung der betroffenen Geräte!
→ mitgelieferte Montageanleitung oder auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

© ifm electronic gmbh | www.ifm.com

6827

Anlaufverhalten der Steuerung

⚠ WARNUNG

Gefahr durch unbeabsichtigtes und gefährliches Anlaufen von Maschinen- oder Anlagenteilen!

► Der Programmierer muss bei der Programmerstellung verhindern, dass nach Auftreten eines Fehlers (z.B. Not-Aus) und der anschließenden Fehlerbeseitigung unbeabsichtigt Maschinen- oder Anlagenteile gefährlich anlaufen können.

► Dazu müssen die in Frage kommenden Ausgänge zusätzlich abgeschaltet und die logischen Zustände mit dem Relaiszustand verknüpft und ausgewertet werden.

Sollte ein zu überwachender Ausgang dauerhaft durchgeschaltet und der Kontakt des Überwachungsrelais verschweißt sein, ist ein Abschalten des jeweiligen Ausganges nicht möglich!

Ein Wiederanlauf kann z.B. verursacht werden durch:
- Spannungswiederkehr nach Spannungsausfall
- Reset nach Watchdog-Ansprechen wegen zu langer Zykluszeit
- Fehlerbeseitigung nach Not-Aus

So erreichen Sie sicheres Verhalten der Steuerung:

► Spannungsversorgung im Applikations-Programm überwachen.

► Im Fehlerfall alle relevanten Ausgänge im Applikations-Programm ausschalten.

► Relaiskontakte, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Applikations-Programm überwachen (Feedback).

► Bei Bedarf in der Applikation sicherstellen, dass verschweißte Relaiskontakte keine gefahrbringenden Bewegungen auslösen oder fortführen können.

2.2 Welche Vorkenntnisse sind notwendig?

215

Das Dokument richtet sich an Personen, die über Kenntnisse der Steuerungstechnik und SPS-Programmierkenntnisse mit IEC 61131-3 verfügen.

Wenn dieses Gerät über eine SPS verfügt, sollten die Personen zusätzlich mit der Software CoDeSys vertraut sein.

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung eines Produkts verursachen kann. Das Dokument enthält Angaben zum korrekten Umgang mit dem Produkt.

Lesen Sie dieses Dokument vor dem Einsatz, damit Sie mit Einsatzbedingungen, Installation und Betrieb vertraut werden. Bewahren Sie das Dokument während der gesamten Einsatzdauer des Gerätes auf.

Befolgen Sie die Sicherheitshinweise.

© ifm electronic gmbh | www.ifm.com

3 Systembeschreibung

Inhalt

Angaben zum Gerät 10
Angaben zur Software 14
Steuerungskonfiguration 15

975

3.1 Angaben zum Gerät

11544

Diese Anleitung beschreibt die Controller-Gerätefamilie **ecomat*mobile*** der **ifm electronic gmbh** mit 32 Bit Mikrocontroller für den mobilen Einsatz:

- BasicController: CR040n
  Laufzeitsystem (Target) V02

3.1.1 Unterschiede in der Gerätefamilie

7988

Die einzelnen Geräte dieser Familie unterscheiden sich wie folgt:

© ifm electronic gmbh | www.ifm.com

CR0401		Grundgerät ohne Abdeckung • Schutzart IP 20 • 12 Eingänge: ▪ IN0...IN3 parametrierbar: digital / analog (12 Bit): - 0...10 V / 0...32 V / 0...20 mA / Ratio - highside/lowside / Diagnose / Frequenz ▪ IN4...IN7 parametrierbar: - digital / analog (12 Bit), diagnosefähig ODER - für Temperatursensoren und Tankgeber ▪ IN8...IN11: digital / analog (12 Bit), diagnosefähig • 8 Ausgänge: ▪ OUT0...OUT3: 1 A highside / PWM ▪ OUT4, OUT5: 2 A highside, diagnosefähig / PWM ▪ OUT6, OUT7: 4 A highside, diagnosefähig / PWM • frei programmierbar mit CoDeSys 2.3 • physikalisch vorhandener SRAM ¹) = 208 KiB • 1 CAN-Schnittstelle (intern) - Protokoll RAW-CAN - für I/O-Erweiterung und Display • 1 CAN-Schnittstelle unterstützt die Protokolle... - RAW-CAN - SAE J1939 - CANopen-Master - CANopen-Slave

© ifm electronic gmbh | www.ifm.com

CR0403		Erweitertes Grundgerät ohne Abdeckung • Schutzart IP 20 • 12 Eingänge: ▪ IN0...IN3 parametrierbar: digital / analog (12 Bit): - 0...10 V / 0...32 V / 0...20 mA / Ratio - highside/lowside / Diagnose / Frequenz ▪ IN4...IN7 parametrierbar: - digital / analog (12 Bit), diagnosefähig ODER - für Temperatursensoren und Tankgeber ▪ IN8...IN11: digital / analog (12 Bit), diagnosefähig • 12 Ausgänge: ▪ OUT0, OUT1: 2 A highside, diagnosefähig / PWM / PWMi ▪ OUT2, OUT3, OUT8, OUT9: 2 A highside, diagnosefähig / PWM ▪ OUT4...OUT7: 1 A highside / PWM ▪ OUT10, OUT11: 4 A highside, diagnosefähig / PWM • frei programmierbar mit CoDeSys 2.3 • physikalisch vorhandener SRAM [1]) = 592 KiB • 1 CAN-Schnittstelle (intern) - Protokoll RAW-CAN - für I/O-Erweiterung und Display • 1 CAN-Schnittstelle unterstützt die Protokolle... - RAW-CAN - SAE J1939 - CANopen-Master - CANopen-Slave

[1]) SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.

© ifm electronic gmbh | www.ifm.com

3.1.2 Zubehör

7989

Zur Gerätefamilie der BasicController gibt es reichhaltiges Zubehör. Beispiele:

CR0421	BasicRelay Verdrahtungs-, Relais- und Sicherungsplattform • Schutzart IP 20 • Steckplätze für 6 Kfz-Relais • Steckplätze für 10 Kfz-Sicherungen
CR0451	BasicDisplay Grafikfähiges, programmierbares Display • Schutzart IP 65 (IP 67 frontseitig) • TFT LCD Farb-Display • frei programmierbar mit CoDeSys 2.3 • 1 CAN-Schnittstelle unterstützt die Protokolle... - RAW-CAN - SAE J1939 - CANopen-Master - CANopen-Slave
EC0401	Gehäuseabdeckung für... • BasicController: CR040n (im CR0402 bereits enthalten) • BasicRelay: CR0421 Abdeckung erhöht zusammen mit beiliegender Kabeldichtung bei BasicController und BasicRelay die Schutzart von IP 20 auf IP 54
EC0402	Gehäuseabdeckung für... • BasicController: CR040n • BasicRelay: CR0421 mit Öffnung für BasicDisplay: CR0451 Abdeckung erhöht zusammen mit beiliegender Kabeldichtung bei BasicController und BasicRelay die Schutzart von IP 20 auf IP 54
EC0403	Montagerahmen für BasicDisplay: CR0451
---	Software "Maintenance Tool" zum Aktualisieren von Firmware, Laufzeitsystem und Applikations-Programm → ecomatmobile-DVD "Software, tools and documentation"

Zubehör zum Artikel finden Sie auf der ifm-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > Artikel-Nr. > [Zubehör]

© ifm electronic gmbh | www.ifm.com

3.2 Angaben zur Software

456

Wir beziehen uns in dieser Anleitung auf CoDeSys Version 2.3.9.33.

Im "Programmierhandbuch CoDeSys 2.3" erhalten Sie weitergehende Informationen über die Nutzung des Programmiersystems "CoDeSys for Automation Alliance". Dieses Handbuch steht auf der **ifm**-Internetseite als kostenloser Download zur Verfügung:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm
→ **ecomat*mobile***-DVD "Software, tools and documentation"

Die Applikations-Software nach IEC 61131-3 kann vom Anwender komfortabel mit dem Programmiersystem CoDeSys selbst erstellt werden. Für den Einsatz dieser Software auf dem PC gelten folgende Mindest-Systemvoraussetzungen:

- CPU Pentium II, 500 MHz
- Arbeitsspeicher (RAM) 128 MB, empfohlen: 256 MB
- Freier Festplattenspeicher (HD) 100 MB
- Betriebssysteme:
  - Windows XP
  - Windows Vista (32 Bit) (! nicht getestet, ohne Gewähr für **ifm**-Produkte)
  - Windows 7 (32/64 Bit) ab CoDeSys 2.3.9.24
- DVD-Laufwerk

Weitere Details zur aktuellen CoDeSys-Software:
DE: → http://www.3s-software.com/index.shtml?de_oem1
UK: → http://www.3s-software.com/index.shtml?en_oem1
FR: → http://www.3s-software.com/index.shtml?fr_oem1

► Der Anwender muss außerdem beachten, welcher Softwarestand (speziell beim R360-Betriebssystem und den Funktionsbibliotheken) zum Einsatz kommt.

! HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Betriebssystems (`ifm_CRnnnn_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CRnnnn_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CRnnnn_Vxxyyzz.LIB`) und
- der weiteren Dateien (→ Kapitel ***Übersicht der verwendeten Dateien und Bibliotheken*** (→ Seite 279)).

CRnnnn	Geräte-Artikelnummer
Vxx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0401`") und die Software-Versionsnummer "xx" (z.B. "`01`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

! Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 1131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

ifm electronic gmbh | www.ifm.com

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CoDeSys nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

ACHTUNG

Gefahr von Datenverlust!

Bei Spannungsausfall während der Datenübertragung können Daten verloren gehen, so dass das Gerät nicht mehr funktionsfähig ist. Reparatur ist nur bei **ifm electronic** möglich.

► Für ständige Spannungsversorgung während der Datenübertragung sorgen!

⚠ WARNUNG

Für die sichere Funktion der Applikations-Programme, die vom Anwender erstellt werden, ist dieser selbst verantwortlich. Bei Bedarf muss er zusätzlich entsprechend der nationalen Vorschriften eine Abnahme durch entsprechende Prüf- und Überwachungsorganisationen durchführen lassen.

3.3 Steuerungskonfiguration

1797

Bei dem Steuerungssystem **ecomat*mobile*** handelt es sich um ein Gerätekonzept für den Serieneinsatz. Das bedeutet, dass die Geräte optimal auf den jeweiligen Einsatzfall konfiguriert werden können.

Die aktuelle Version der **ecomat*mobile***-Software kann über www.ifm.com aus dem Internet geladen werden.

→ ***Target einrichten*** (→ Seite 17).

Ob bestimmte in der Dokumentation beschriebene Funktionen, Hardwareoptionen, Ein- und Ausgänge in der betreffenden Hardware verfügbar sind, muss in jedem Fall vor Einsatz der Geräte überprüft werden.

© ifm electronic gmbh | www.ifm.com

4 Konfigurationen

Inhalt

Programmiersystem einrichten....16
Funktionskonfiguration der Ein- und Ausgänge....22
Hinweise zur Anschlussbelegung....33

3615

4.1 Programmiersystem einrichten

Inhalt

Programmiersystem manuell einrichten....16
Programmiersystem über ifm-Templates einrichten....21

3968

4.1.1 Programmiersystem manuell einrichten

Inhalt

Target einrichten....17
Steuerungskonfiguration aktivieren (z.B. CR1080)....18
CAN-Deklaration (z.B. CR1080)....20

3963

© ifm electronic gmbh

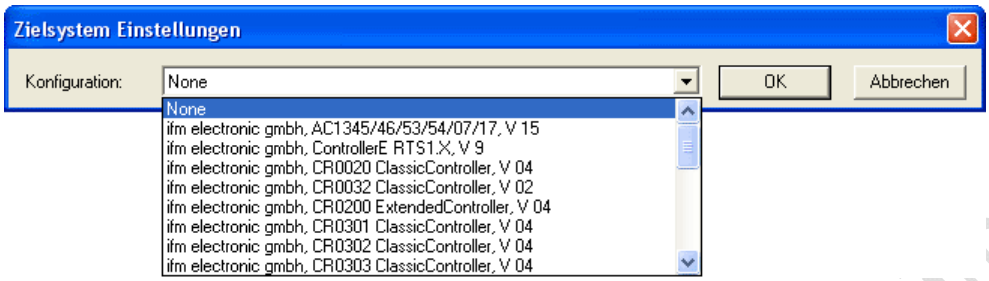
Target einrichten

2687

11379

Beim Erstellen eines neuen Projektes in CoDeSys muss die dem Gerät entsprechende Target-Datei geladen werden.

► Die gewünschte Target-Datei im Dialogfenster wählen (→ Screenshot).


Zielsystem Einstellungen
Konfiguration:
None
OK
Abbrechen
None
ifm electronic gmbh, AC1345/46/53/54/07/17, V 15
ifm electronic gmbh, ControllerE RTS1.X, V 9
ifm electronic gmbh, CR0020 ClassicController, V 04
ifm electronic gmbh, CR0032 ClassicController, V 02
ifm electronic gmbh, CR0200 ExtendedController, V 04
ifm electronic gmbh, CR0301 ClassicController, V 04
ifm electronic gmbh, CR0302 ClassicController, V 04
ifm electronic gmbh, CR0303 ClassicController, V 04

Grafik: Zielsystem Einstellungen (Beispiel)

> Die Target-Datei stellt für das Programmiersystem die Schnittstelle zur Hardware her.

> Gleichzeitig mit Wahl des Targets werden automatisch einige wichtige Bibliotheken und die Steuerungskonfiguration geladen.

► Bei Bedarf geladene Bibliotheken wieder entfernen oder durch weitere Bibliotheken ergänzen.

► Immer die passende Geräte-Bibliothek `ifm_CRnnnn_Vxxyyzz.LIB` manuell ergänzen!

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Betriebssystems (`ifm_CRnnnn_Vxxyyzz.H86` / `ifm_CRnnnn_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CRnnnn_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CRnnnn_Vxxyyzz.LIB`) und
- der weiteren Dateien (→ Kapitel ***Übersicht der verwendeten Dateien und Bibliotheken*** (→ Seite 279)).

CRnnnn	Geräte-Artikelnummer
Vxx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0032`") und die Software-Versionsnummer "xx" (z.B. "`02`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 1131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CoDeSys nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

ifm Lay, ifm gmbh | www.ifm.com

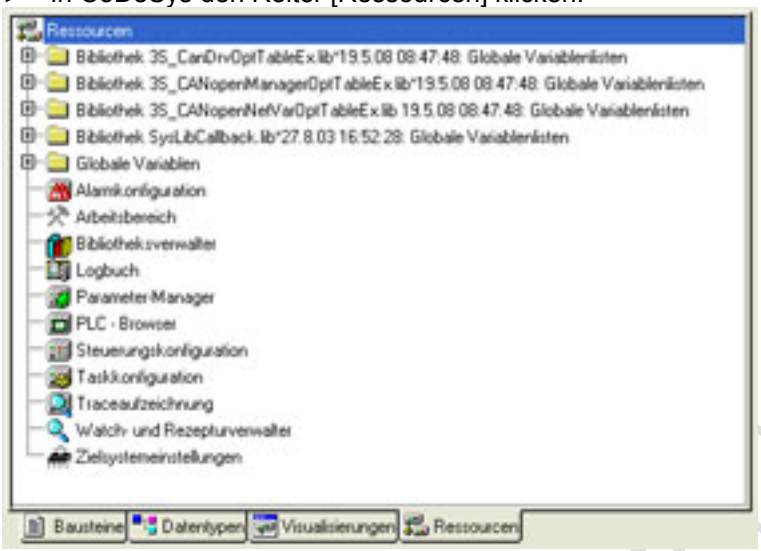
Steuerungskonfiguration aktivieren (z.B. CR1080)

10079

Bei der Konfiguration des Programmiersystems (→ vorheriger Abschnitt) erfolgte automatisch auch die Steuerungskonfiguration.

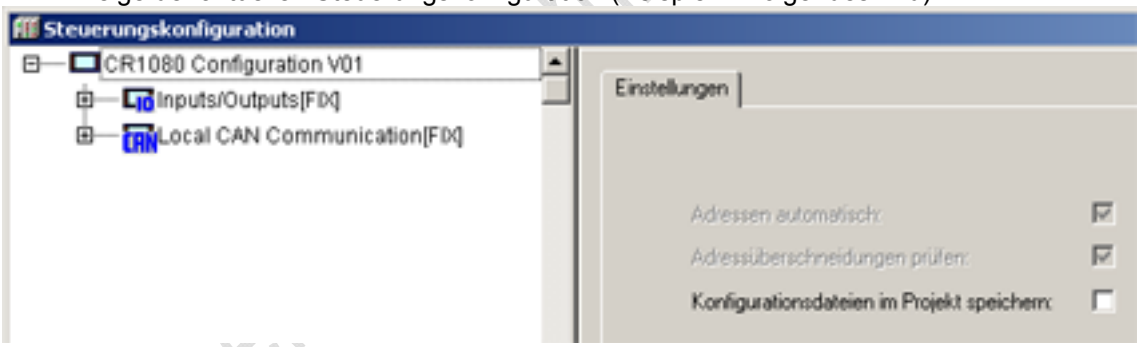
Den Punkt [Steuerungskonfiguration] erreicht man über den Reiter [Ressourcen]. Über einen Doppelklick auf den Punkt [Steuerungskonfiguration] öffnet sich das entsprechende Fenster.

► In CoDeSys den Reiter [Ressourcen] klicken:



► In der linken Spalte Doppelklick auf [Steuerungskonfiguration]

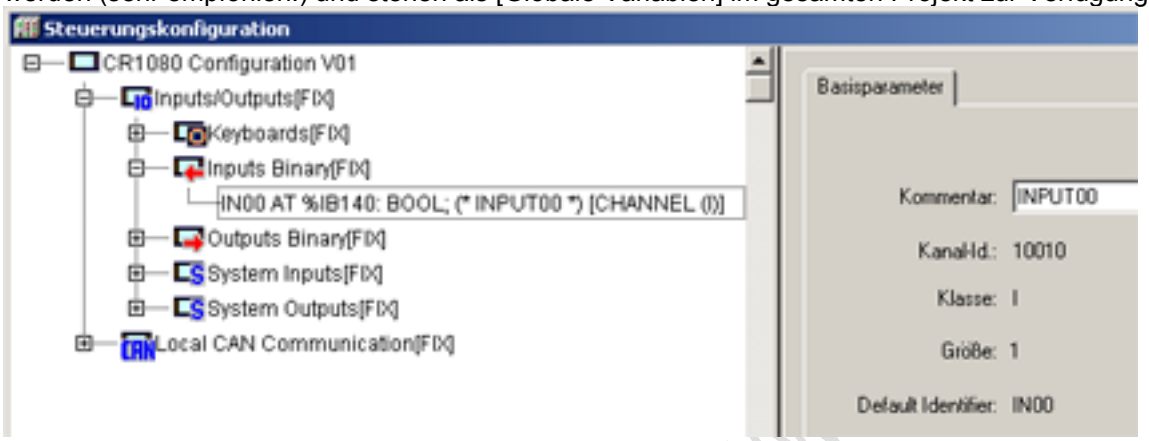
> Anzeige der aktuellen Steuerungskonfiguration (Beispiel → folgendes Bild):



© ifm electronic www.ifm.com

Durch die Konfiguration erhält der Anwender in der Programmumgebung Folgendes verfügbar:

- alle wichtigen System- und Fehlermerker
  Je nach Anwendung und Applikations-Programm müssen diese Merker bearbeitet und ausgewertet werden. Der Zugriff erfolgt über deren symbolischen Namen.
- die Struktur der Ein- und Ausgänge
  Diese können im Fenster [Steuerungskonfiguration] (→ Bild unten) direkt symbolisch bezeichnet werden (sehr empfohlen!) und stehen als [Globale Variablen] im gesamten Projekt zur Verfügung.



© ifm electronic gmbh | ww

CAN-Deklaration (z.B. CR1080)

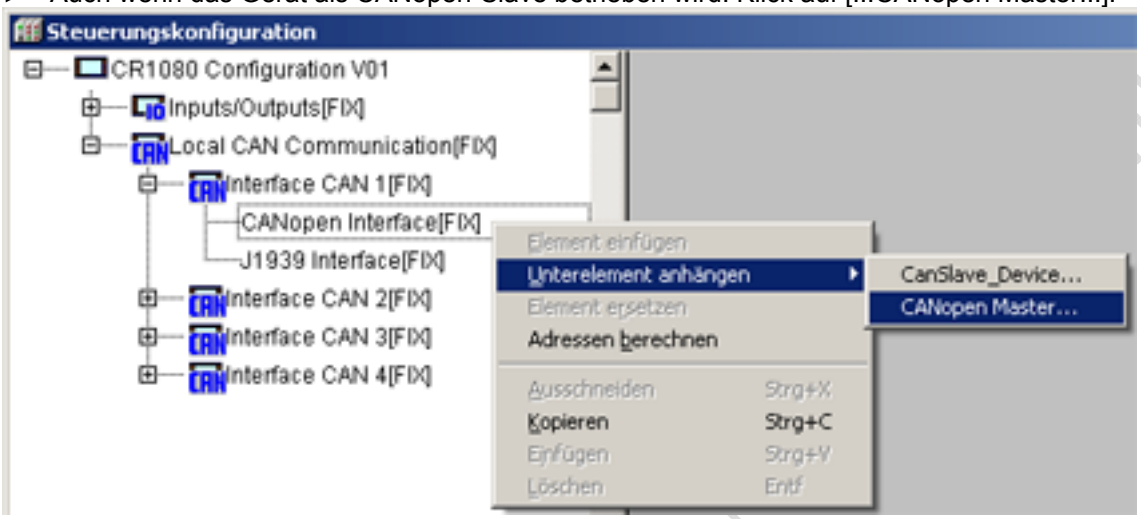
10080

In der Steuerungskonfiguration müssen Sie nun die CAN-Schnittstelle(n) deklarieren.

► Rechtsklick auf [CANopen Interface [FIX]] der gewünschten CAN-Schnittstelle.

► Klick auf [Unterelement anhängen].

► Auch wenn das Gerät als CANopen-Slave betrieben wird: Klick auf [...CANopen Master...]:

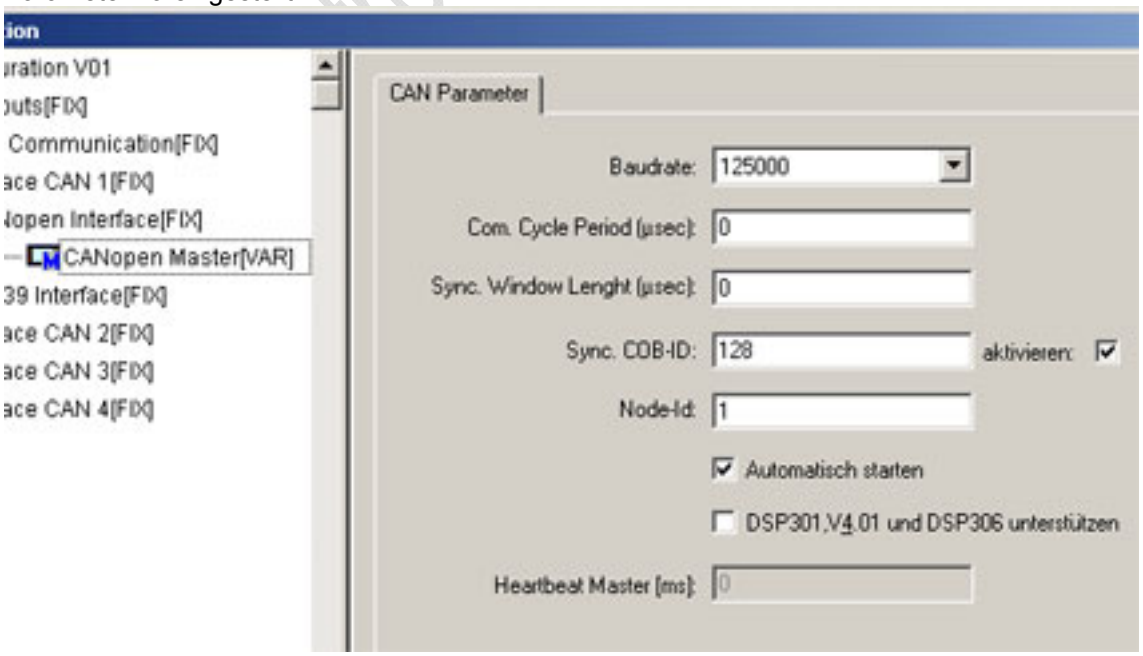


Info

Wenn das Gerät als Slave betrieben wird, wäre die Auswahl [...CANopen Slave...] ebenfalls möglich.
→ Kapitel ***CANopen-Slave*** (→ Seite 150)

Bei der insgesamt einfacheren Konfiguration als Master können auch alle CAN-Layer2- und Netzwerkvariablen-Funktionen genutzt werden.

> Die CAN-Parameter der Steuerungskonfiguration erscheinen. Hier sind bereits einige CAN-Parameter voreingestellt:



► Wenn das Gerät via Netzwerkvariablen oder CAN_RX / CAN_TX auf CAN-Layer 2 oder als Slave betrieben wird:
Prüfen, ob für das Gerät die richtige Baudrate eingestellt ist (Baudrate muss für alle Teilnehmer identisch sein).

► Wenn das Gerät als CANopen-Master betrieben wird:
Alle Parameter-Einstellungen prüfen.

► Das Fenster [Steuerungskonfiguration] schließen.

► Mit Menü [Datei] > [speichern unter...] dem Projekt einen sinnvollen Namen geben und das Projekt im gewünschten Verzeichnis speichern.

► Im Applikations-Program für jede CAN-Schnittstelle immer eine eigene Instanz des FBs ***CANOPEN_ENABLE*** (→ Seite 164) anlegen!

4.1.2 Programmiersystem über ifm-Templates einrichten

11649

Wir erstellen und aktualisieren ständig Templates und Demos, um Ihnen das Einrichten des Programmiersystems zu erleichtern. Sie finden diese kleinen CoDeSys-Projekte auf unserer DVD:
→ **ecomat*mobile***-DVD "Software, tools and documentation"

Eine Beschreibung zu den Templates und Demos finden Sie in der Anleitung "ifm ecomatmobile Templates und Demos" auf unserer Homepage:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm

© ifm electronic gmbh | www.ifm.com

4.2 Funktionskonfiguration der Ein- und Ausgänge

Inhalt

Eingänge konfigurieren 23
Ausgänge konfigurieren 29

7995

Bei einigen Geräten der Controller-Familie **ecomatmobile** sind bei den Ein- und Ausgängen zusätzliche Diagnosefunktionen aktivierbar. Damit kann das jeweilige Ein- und Ausgangssignal überwacht werden und im Fehlerfall kann das Applikations-Programm darauf reagieren.

Je nach Ein- und Ausgang müssen bei der Nutzung der Diagnose bestimmte Randbedingungen beachtet werden:

- Anhand des Datenblattes muss überprüft werden, ob das eingesetzte Gerät die beschriebenen Ein- und Ausgangsgruppen zur Verfügung stellt.
- Zur Konfiguration der Ein- und Ausgänge sind in den Gerätebibliotheken (z.B. `ifm_CR0032_Vxxyyzz.LIB`) Konstanten vordefiniert (z.B. IN_DIGITAL_H).
  Ausführliche Angaben → ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 262, → Seite 263).

© ifm electronic gmbh | www.ifm.com

4.2.1 Eingänge konfigurieren

Inhalt

Sicherheitshinweise....23
Diagnose der Eingänge aktivieren....23
Digitaleingänge....24
Schnelle Eingänge....24
Analogeingänge....25
Widerstandsmessung....26
Eingangsgruppe I0 (%IX0.0...%IX0.3)....27
Eingangsgruppe I1 (%IX0.4...%IX0.7)....28
Eingangsgruppe I2 (%IX0.8...%IX0.11)....28

3973

Sicherheitshinweise

7348

Beim Einsatz von nichtelektronischen Schaltern Folgendes beachten:

Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

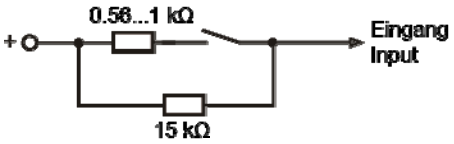
► **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
Beispiel: 32 V / 500 mA = 64 Ohm

► Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
Beispiel:
RE = 3 000 Ohm
⇨ max. Vorwiderstand = 150 Ohm

Diagnose der Eingänge aktivieren

7352

Soll die Diagnose genutzt werden, muss diese zusätzlich aktiviert werden.

Diagnose bei nichtelektronischen Schaltern: Um die Eingangssignale von nichtelektronischen Schaltern zu überwachen, müssen diese mit einer zusätzlichen Widerstandsbeschaltung versehen werden.	 Grafik: Nichtelektronische Schalter

An diesen Eingängen können diagnosefähige Sensoren nach NAMUR verwendet werden. Eine zusätzliche Widerstandsbeschaltung ist dann nicht erforderlich.

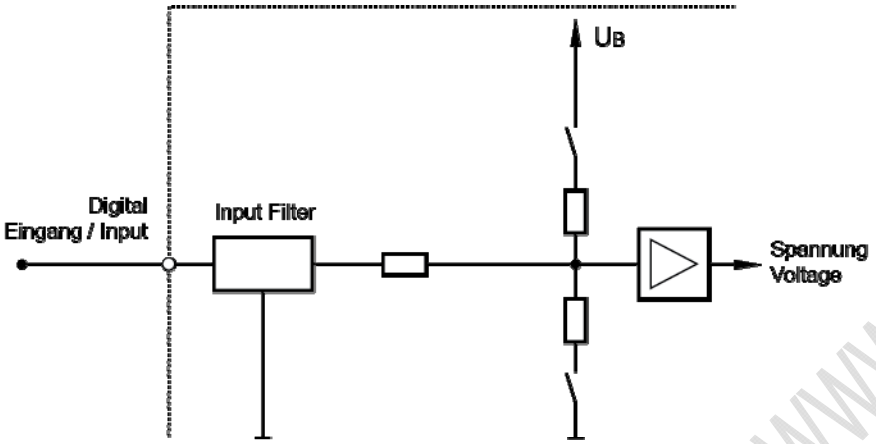
> Der FB ***INPUT*** (→ Seite 212) liefert die Diagnosemeldungen der Eingänge an seinem Ausgang RESULT.

© ifm electronic gmbh | www.ifm.com

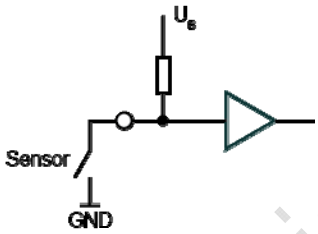
Digitaleingänge

1015

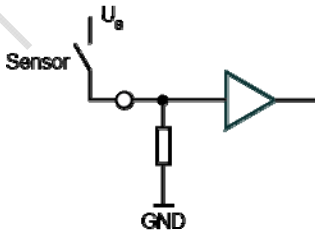
Je nach Controller können auch die Digitaleingänge unterschiedlich konfiguriert werden. Neben den Schutzmechanismen gegen Störungen werden die Digitaleingänge intern über eine Analogstufe ausgewertet. Das ermöglicht die Diagnose der Eingangssignale. In der Applikations-Software steht das Schaltsignal aber direkt als Bit-Information zur Verfügung. Bei einem Teil dieser Eingänge (CRnn32: bei allen Eingängen) kann auch das Potential gewählt werden, gegen das geschaltet wird.



Grafik: Prinzipschaltung High-/Lowside Eingang für negative und positive Gebersignale



Highside Eingang für negatives Gebersignal



Lowside Eingang für positives Gebersignal

Schnelle Eingänge

8292

Zusätzlich verfügen die Controller über 4 schnelle Zähl-/Impulseingänge für eine Eingangsfrequenz bis 30 kHz (→ Datenblatt). Werden z.B. mechanische Schalter an diesen Eingängen angeschlossen, kann es durch Kontaktprellen zu Fehlsignalen in der Steuerung kommen.

Z.B. folgende Funktionsblöcke können Sie hier sinnvoll einsetzen:

- Schneller Zähler mit ***FASTCOUNT*** (→ Seite 216).
- Periodendauermessung mit ***PERIOD*** (→ Seite 221).
- Drehgeber auswerten an je zwei schnellen Eingängen mit ***INC_ENCODER*** (→ Seite 218).

Bei Einsatz dieser Bausteine werden automatisch die dort parametrierten Ein-/Ausgänge konfiguriert. Der Programmierer der Applikation ist hiervon entlastet.

© ifm electronic gmbh | www.ifm.com

Analogeingänge

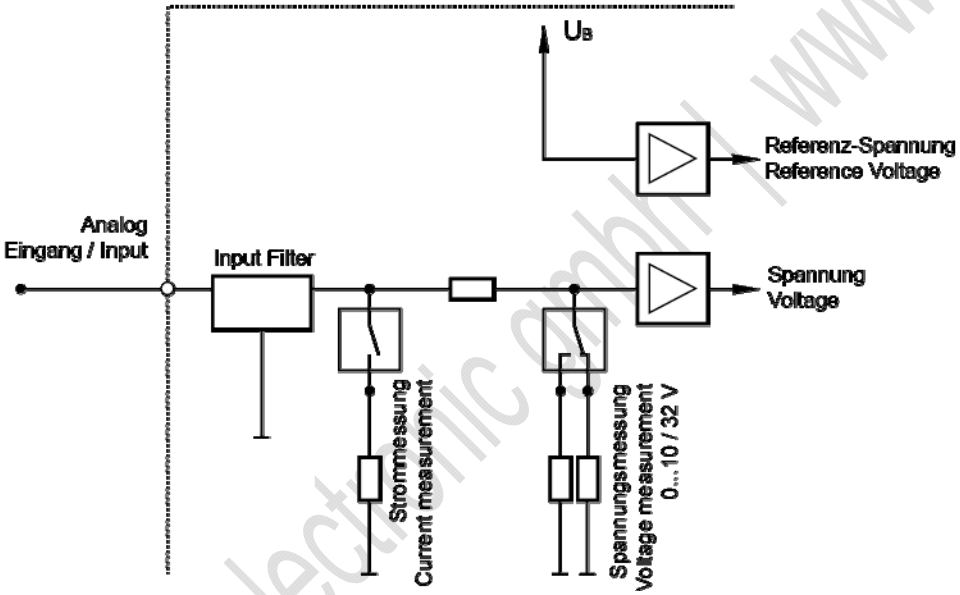
2426

Die Eingänge können auch als Analogeingänge über das Applikationsprogramm konfiguriert werden. Der Messbereich kann zwischen folgenden Bereichen umgeschaltet werden:

- Stromeingang 0...20 mA
- Spannungseingang 0...10 V
- Spannungseingang 0...32 V

Wird in der Betriebsart "0...32 V" die Versorgungsspannung zurückgelesen, kann die Messung auch ratiometrisch erfolgen. Das bedeutet, ohne zusätzliche Referenzspannung können Potentiometer oder Joysticks ausgewertet werden. Ein Schwanken der Versorgungsspannung hat auf diesen Messwert dann keinen Einfluss.

Bei ratiometrischer Messung sollten die angeschlossenen Sensoren mit der gleichen Spannungsquelle wie der Controller versorgt werden. Dadurch werden Fehlmessungen durch Spannungsverschiebungen vermieden.



Grafik: Prinzipschaltung der Analogeingänge

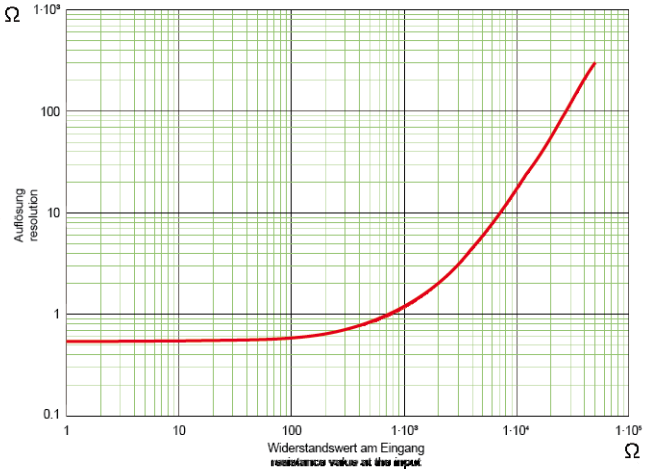
© ifm electronic gmbH | www.ifm.com

Widerstandsmessung

9773

Gilt nur für folgende Geräte:
- BasicController: CR040n
- ClassicController: CR0033
- ExtendedController: CR0233

Bei diesem Gerät ist die Auflösung nicht linear abhängig vom Widerstandswert, → Grafik:



Um wieviel Ohm ändert sich der Messwert, wenn sich das Signal des A/D-Wandlers am Eingang um 1 ändert? Beispiele:

- Im Bereich 1...100 Ohm beträgt die Auflösung 0,5 Ohm.
- Im Bereich bei 3 kOhm beträgt die Auflösung ca. 3 Ohm.
- Im Bereich bei 20 kOhm beträgt die Auflösung ca. 55 Ohm.

© ifm electronic gmbh | www.ifm.com

Eingangsgruppe I0 (%IX0.0...%IX0.3)

8293

Diese Eingänge können wahlweise (jeder Eingang separat konfigurierbar) wie folgt genutzt werden:

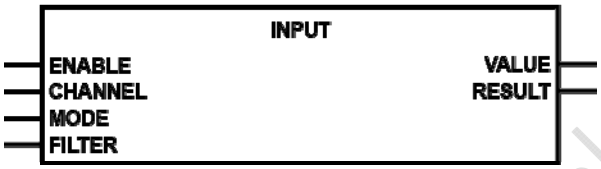
- digitaler Lowside-Eingang für positives Gebersignal
- digitaler Highside-Eingang für negatives Gebersignal
- schneller Eingang
- analoger Eingang

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen. Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

Alle Eingänge zeigen das gleiche Verhalten bei Funktion und Diagnose.

Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (21,7 mA) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet und der Ausgang RESULT des Funktionsblock INPUT auf den entsprechenden Wert gesetzt. Wird der Grenzwert wieder unterschritten, schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

Die Konfiguration der Eingänge erfolgt über die Applikations-Software. ***INPUT*** (→ Seite 212) konfiguriert die Betriebsart des gewählten Analogkanals über den Eingang MODE (→ folgendes Beispiel).



© ifm electronic gmbh | www.ifm.com

Eingangsgruppe I1 (%IX0.4...%IX0.7)

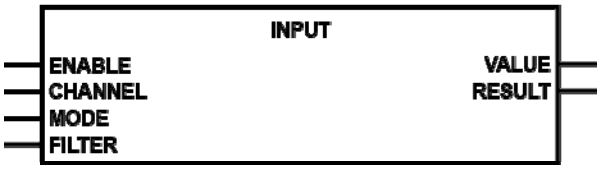
8294

Diese Eingänge können wahlweise (jeder Eingang separat konfigurierbar) wie folgt genutzt werden:

- digitaler Lowside-Eingang für positives Gebersignal
- Eingang für Temperatursensoren oder Tankgeber

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

Die Konfiguration der Eingänge erfolgt über die Applikations-Software. ***INPUT*** (→ Seite 212) konfiguriert die Betriebsart des gewählten Analogkanals über den Eingang MODE (→ folgendes Beispiel).



Eingangsgruppe I2 (%IX0.8...%IX0.11)

8295

Diese Eingänge können wie folgt genutzt werden:

- digitaler Lowside-Eingang für positives Gebersignal

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

© ifm electronic gmbh | www.ifm.com

4.2.2 Ausgänge konfigurieren

Inhalt

Digital- und PWM-Ausgänge 29
Diagnose der Ausgänge konfigurieren 30
Ausgangsgruppe 1 30
Ausgangsgruppe 2 31
Ausgangsgruppe 3 32

3976

Digital- und PWM-Ausgänge

9226

Bei den Geräte-Ausgängen können zwei Typen unterschieden werden:

- Highside-Digitalausgänge mit und ohne Diagnosefunktion und zusätzlichem PWM-Modus
- Lowside-Digitalausgänge ohne Diagnosefunktion

Einige PWM-Ausgänge können mit und ohne Stromregelfunktion betrieben werden. Stromgeregelte PWM-Ausgänge werden überwiegend zur Ansteuerung von proportionalen Hydraulikfunktionen genutzt.

⚠ WARNUNG

Sach- oder Körperschäden möglich durch Fehlfunktionen!

Ausgänge, die im PWM-Modus betrieben werden, unterstützen keine Diagnosefunktionen und es werden keine ERROR-Merker gesetzt. Dies ist bedingt durch den Aufbau der Ausgänge.

Der Überlastschutz OUT_OVERLOAD_PROTECTION steht in diesem Modus nicht zur Verfügung!

⚠ WARNUNG

Sach- oder Körperschäden möglich durch Fehlfunktionen!

► Für sicherheitsrelevante Applikationen sind vorzugsweise die Ausgänge mit Rücklesefunktion (diagnosefähige Ausgänge) einzusetzen und im Applikations-Programm zu überwachen.

HINWEIS

Wenn die Diagnose im Fehlerfall einen Ausgang abschaltet:

> Der im Applikations-Programm für diesen Ausgang erzeugte logische Zustand ändert sich dadurch NICHT.

> Der Ausgang ist deaktiviert.

Abhilfe:

► Den Fehler beseitigen.

► Den Ausgang auch im Applikations-Programm abschalten.

> Der Ausgang kann wieder aktiv verwendet werden.

© ifm electronic gmbh | www.ifm.com

Diagnose der Ausgänge konfigurieren

8301

Soll die Diagnose genutzt werden, muss diese zusätzlich aktiviert werden.

► Am Eingang MODE des FB OUTPUT die Betriebsart des Ausgangs einstellen.

> Der FB ***OUTPUT*** (→ Seite 224) liefert die Diagnosemeldungen der Ausgänge an seinem Ausgang RESULT.

Ausgangsgruppe 1

8300

Gerät	Gruppe	IEC-Adressen	symbolische Adressen
CR0401	---	---	---
CR0403	Q0	%QB0...%QB3	OUT00…OUT03

Diese Ausgänge bieten mehrere Funktionsmöglichkeiten:

- digitaler Ausgang Highside
- analoger stromgeregelter Ausgang (nur OUT00, OUT01)
- analoger Ausgang mit Pulsweitenmodulation

Werden die Ausgänge nicht als PWM-Ausgänge genutzt, wird die Diagnose über die integrierte Spannungsmessung der Ausgangspegel realisiert.

Mit ***OUTPUT*** (→ Seite 224) können die Lastströme angezeigt werden.

Die Ausgänge im PWM-Modus unterstützen keine Diagnosefunktionen.

HINWEIS

Wenn die Diagnose im Fehlerfall einen Ausgang abschaltet:

> Der im Applikations-Programm für diesen Ausgang erzeugte logische Zustand ändert sich dadurch NICHT.

> Der Ausgang ist deaktiviert.

Abhilfe:

► Den Fehler beseitigen.

► Den Ausgang auch im Applikations-Programm abschalten.

> Der Ausgang kann wieder aktiv verwendet werden.

Bei der Nutzung als Digitalausgang erfolgt die Konfiguration jedes Ausgangs mit dem Eingang MODE des FB ***OUTPUT*** (→ Seite 224). Soll die Diagnose genutzt werden, muss der entsprechende Modus aktiviert werden.

© ifm electronic gmbh | www.ifm.com

HINWEIS

Um die internen Messwiderstände zu schützen, sollte im FB OUTPUT für die Ausgänge mit Stromerfassung mit MODE = PROTECTION der Schutz aktiviert sein. Es besteht Schutz ab 1,125x Nennstrom. Die Funktion wird NICHT im reinen PWM-Modus unterstützt und kann bei Bedarf abgeschaltet werden.

Zu den Grenzwerten unbedingt das Datenblatt beachten!

Abhängig von der Umgebungstemperatur kann ab einem bestimmten Kurzschlussstrom ein Kurzschluss eventuell nicht mehr zuverlässig erkannt werden, da die Ausgangstreiber sich zum Schutz vor Zerstörung selbsttätig zeitweise deaktivieren.

Die Kurzschlusserkennung ist aktiv, wenn der Ausgang EINgeschaltet ist.
Die Leiterbrucherkennung ist aktiv, wenn der Ausgang AUSgeschaltet ist.

> Der FB ***OUTPUT*** (→ Seite 224) liefert die Diagnosemeldungen der Ausgänge an seinem Ausgang RESULT.

Ausgangsgruppe 2

8305

Gerät	Gruppe	IEC-Adressen	symbolische Adressen
CR0401	Q0	%QB0...%QB3	OUT00…OUT03
CR0403	Q1	%QB4...%QB7	OUT04…OUT07

Diese Ausgänge bieten mehrere Funktionsmöglichkeiten:

- digitaler Ausgang Highside
- analoger Ausgang mit Pulsweitenmodulation

Sie sind kurzschluss- und überlastfest. Die Ausgänge unterstützen keine Diagnosefunktionen.

Wird ein Ausgang im Fehlerfall hardwaremäßig abgeschaltet, ändert sich der durch das Applikations-Programm erzeugte logische Zustand dadurch nicht.

© ifm electronic gmbh | www.ifm.com

Ausgangsgruppe 3

8306

Gerät	Gruppe	IEC-Adressen	symbolische Adressen
CR0401	Q1	%QB4...%QB7	OUT04…OUT07
CR0403	Q2	%QB8...%QB11	OUT08…OUT11

Diese Ausgänge bieten mehrere Funktionsmöglichkeiten:

- digitaler Ausgang Highside
- analoger Ausgang mit Pulsweitenmodulation

Werden die Ausgänge nicht als PWM-Ausgänge genutzt, wird die Diagnose über die integrierte Spannungsmessung der Ausgangspegel realisiert.

Die Ausgänge im PWM-Modus unterstützen keine Diagnosefunktionen.

HINWEIS

Wenn die Diagnose im Fehlerfall einen Ausgang abschaltet:

> Der im Applikations-Programm für diesen Ausgang erzeugte logische Zustand ändert sich dadurch NICHT.

> Der Ausgang ist deaktiviert.

Abhilfe:

► Den Fehler beseitigen.

► Den Ausgang auch im Applikations-Programm abschalten.

> Der Ausgang kann wieder aktiv verwendet werden.

Bei der Nutzung als Digitalausgang erfolgt die Konfiguration jedes Ausgangs mit dem Eingang MODE des FB ***OUTPUT*** (→ Seite 224). Soll die Diagnose genutzt werden, muss der entsprechende Modus aktiviert werden.

> Der FB ***OUTPUT*** (→ Seite 224) liefert die Diagnosemeldungen der Ausgänge an seinem Ausgang RESULT.

© ifm electronic gmbh | www.ifm.com

4.3 Hinweise zur Anschlussbelegung

1426

Die Anschlussbelegungen (→ Montageanleitungen der Geräte, Kapitel "Anschlussbelegung") beschreiben die Standard-Gerätekonfigurationen. Die Anschlussbelegung dient der Zuordnung der Ein- und Ausgangskanäle zu den IEC-Adressen und den Geräteanschlussklemmen.

Beispiele:

12 GND_A

12	Klemmennummer
GND_A	Klemmenbezeichnung

30 %IX0.7 BL

30	Klemmennummer
%IX0.7	IEC-Adresse für einen binären Eingang
BL	hardwaremäßige Ausführung des Eingangs, hier: **B**inär **L**ow-Side

47 %QX0.3 BH/PH

47	Klemmennummer
%QX0.3	IEC-Adresse für einen binären Ausgang
BH/PH	Hardwaremäßige Ausführung des Ausgangs, hier: **B**inär-**H**igh-Side oder **P**WM-**H**igh-Side

Die einzelnen Kürzel haben folgende Bedeutung:

A	Analog-Eingang
BH	Binärer Eingang/Ausgang, High-Side
BL	Binärer Eingang/Ausgang, Low-Side
CYL	Eingang Periodendauermessung
ENC	Eingang Drehgebersignale
FRQ	Frequenzeingang
H-Bridge	Ausgang mit H-Brücken-Funktion
PWM	**P**uls**w**eiten-**m**oduliertes Signal
PWM_I	PWM-Ausgang mit Strommessung
IH	Impuls-/Zählereingang, High-Side
IL	Impuls-/Zählereingang, Low-Side
R	Rücklesekanal für einen Ausgang

Zuordnung der Ein-/Ausgangskanäle:

Je nach Gerätekonfiguration steht an einer Geräteklemme ein Eingang und/oder ein Ausgang zur Verfügung (→ Katalog, Montageanleitung oder Datenblatt des jeweiligen Gerätes).

© ifm electronic gmbh | www.ifm.com

Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

► **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
Beispiel: 32 V / 500 mA = 64 Ohm

► Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
Beispiel:
RE = 3 000 Ohm
⇨ max. Vorwiderstand = 150 Ohm

© ifm electronic gmbh | www.ifm.com

5 Betriebszustände und Betriebssystem

Inhalt

Betriebszustände....35
Status-LED....38
Betriebssystem....39

1074

5.1 Betriebszustände

Inhalt

INIT-Zustand (Reset)....35
RUN-Zustand....36
STOP-Zustand....36
ERROR-Zustand....37
FATAL-ERROR-Zustand....37

1075

Nach Anlegen der Versorgungsspannung kann sich der ecomatmobile-Controller in einem von fünf möglichen Betriebszuständen befinden:

5.1.1 INIT-Zustand (Reset)

1076

Dieser Zustand wird nach jedem Power-On-Reset durchlaufen:

> Das Betriebssystem wird initialisiert.

> Verschiedene Checks werden durchgeführt, z.B. Warten auf gültige Versorungsspannung.

> Dieser nur temporäre Zustand wird vom RUN- oder STOP-Zustand abgelöst.

> Die LED leuchtet gelb.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- FATAL ERROR
- STOP

© ifm electronic gmbh | www.ifm

5.1.2 RUN-Zustand

8287

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Aus dem INIT-Zustand (Autostart), wenn folgende Bedingungen erfüllt sind:
  - Die Betriebsspannung hat einen Mindestwert erreicht. UND:
  - Die IEC-Applikation läuft regulär zyklisch.
- Aus dem STOP-Zustand:
  - durch das CoDeSys-RUN-Kommando.
  - Die Betriebsspannung hat einen Mindestwert erreicht oder überschritten.

Im RUN-Zustand:

> Das Laufzeitsystem läuft.

> Das Betriebssystem läuft.

> Das Applikations-Programm läuft.

> Die LED blinkt grün mit 2 Hz.

> Die LED kann vom Applikations-Programm abweichend gesteuert werden.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- STOP
- ERROR
- FATAL ERROR
- INIT (nach Power-On-Reset)

5.1.3 STOP-Zustand

8288

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Aus dem INIT-Zustand, wenn kein Applikations-Programm geladen ist.
- Aus dem RUN-Zustand, wenn folgende Bedingung erfüllt ist:
  - STOP-Kommando kommt über die CoDeSys-Schnittstelle.

Im STOP-Zustand:

> Die Ausgänge des Geräts sind abgeschaltet.

> Das Abarbeiten des Applikations-Programms ist angehalten.

> Die LED leuchtet grün.

Wenn kein Betriebs- / Laufzeitsystem geladen ist: Die LED blinkt grün mit 5 Hz.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- ERROR
- FATAL ERROR
- INIT (nach Power-On-Reset)

© ifm electronic gmbh | www.ifm.com

5.1.4 ERROR-Zustand

8290

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Bei zu geringer Versorgungsspannung.

Im ERROR-Zustand:

> Die Ausgänge des Geräts sind ausgeschaltet.
> Das Abarbeiten des Applikations-Programms ist angehalten.
> Systemparameter werden gespeichert.
> Die LED blinkt rot mit 5 Hz.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- STOP
- FATAL ERROR
- INIT (nach Power-On-Reset)

5.1.5 FATAL-ERROR-Zustand

8289

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Speicherfehler (RAM / Flash)
- Ausnahmezustand
- Betriebssystem-Fehler

Im FATAL-ERROR-Zustand:

> Die Ausgänge des Geräts sind abgeschaltet.
> Das Applikations-Programm ist beendet.
> Das Laufzeitsystem ist beendet.
> Die LED leuchtet rot.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- INIT (nach Power-On-Reset)

© ifm electronic gmbh | www.ifm.com

5.2 Status-LED

7998

Die Betriebszustände werden durch die integrierte Status-LED (Voreinstellung) angezeigt.

LED-Farbe	Blinkfrequenz	Beschreibung
aus	konstant aus	keine Betriebsspannung
Rot + Grün	kurzzeitig ein	INIT-Zustand, Reset-Checks
Grün	5 Hz	kein Betriebssystem geladen
Grün	2 Hz	RUN-Zustand: Applikation läuft
Grün	konstant ein	STOP-Zustand: Applikation angehalten
Rot	5 Hz	STOP-Zustand mit Fehler: Applikation angehalten Ursache: Unterspannung
Rot	10 Hz	STOP-Zustand mit Fehler: Applikation angehalten Ursache: Laufzeitüberschreitung der Applikation oder Visualisierung: ▶ Applikation löschen! ▶ PowerOn-Reset ▶ Applikation neu ins Gerät laden
Rot	konstant ein	FATAL-ERROR: Applikation angehalten Ursache: Software-Watchdog ist ausgefallen ▶ PowerOn-Reset Wenn ohne Erfolg: ▶ Goto Bootloader ▶ PowerOn-Reset ▶ BasicSystem neu ins Gerät laden ▶ Applikation neu ins Gerät laden Wenn ohne Erfolg: ▶ Hardware-Fehler: Gerät an ifm einsenden!

Die Betriebszustände STOP und RUN können vom Programmiersystem geändert werden.

Bei diesem Controller kann die Status-LED auch im Applikations-Programm gesetzt werden. Dazu dient ***SET_LED*** (→ Seite 257).

Der Einsatz des LED-Bausteins im Applikations-Programm ersetzt im RUN -Zustand die System-Voreinstellung der LED.

© ifm electronic gmbh | www.ifm.com

5.3 Betriebssystem

8000

5.3.1 Betriebssystem laden

8486

Im Auslieferungszustand ist im Normalfall kein Betriebssystem im Gerät geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Boot-Lader aktiv. Dieser stellt die minimalen Funktionen für den Betriebssystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. RS232, CAN).

Der Betriebssystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Applikations-Programm kann anschließend (auch mehrfach) in das Gerät geladen werden, ohne das Betriebssystem zu beeinflussen. Vorteil:

Das Betriebssystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm

ACHTUNG

Gefahr von Datenverlust!

Bei Spannungsausfall während der Datenübertragung können Daten verloren gehen, so dass das Gerät nicht mehr funktionsfähig ist. Reparatur ist nur bei **ifm electronic** möglich.

► Für ständige Spannungsversorgung während der Datenübertragung sorgen!

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Betriebssystems (`ifm_CRnnnn_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CRnnnn_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CRnnnn_Vxxyyzz.LIB`) und
- der weiteren Dateien (→ Kapitel ***Übersicht der verwendeten Dateien und Bibliotheken*** (→ Seite 279)).

CRnnnn	Geräte-Artikelnummer
Vxx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0401`") und die Software-Versionsnummer "xx" (z.B. "`01`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 1131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CoDeSys nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

© ifm electronic gmbh | www.ifm.com

Das Betriebssystem wird mit dem eigenständigen Programm "Maintenance Tool" in das Gerät übertragen. (Das Programm befindet sich auf der ecomat*mobile*-DVD "Software, tools and documentation" oder kann bei Bedarf von der ifm-Homepage heruntergeladen werden)

Das Applikations-Programm wird im Normalfall über das Programmiersystem in das Gerät geladen. Es kann aber ebenfalls mit dem "Maintenance Tool" geladen werden, wenn es zuvor aus dem Gerät ausgelesen wurde (→ Upload).

© ifm electronic gmbh | www.ifm.com

6 Begrenzungen und Programmierhinweise

Inhalt

Leistungsgrenzen des Geräts 41
Programmierhinweise für CoDeSys-Projekte.....44

3055

Hier zeigen wir Ihnen die Grenzen des Geräts und helfen Ihnen mit Programmierhinweisen.

6.1 Leistungsgrenzen des Geräts

7358

Beachten Sie die Grenzen des Geräts! → Datenblatt

6.1.1 CPU-Frequenzen

11643

► Beachten Sie, welche CPU in dem eingesetzten Gerät verwendet wird:

Geräte-Familie / Artikel-Nr.	CPU-Frequenz [MHz]
BasicController: CR040n	50
BasicDisplay: CR0451	50

Je höher die CPU-Frequenz, desto größer ist die Leistungsfähigkeit für den gleichzeitigen Einsatz von komplexen Bausteinen.

© ifm electronic gmbh | www.ifm.com

6.1.2 Verhalten des Watchdog

11644

Bei (fast) allen programmierbaren ecomatmobile-Geräten wird CoDeSys-intern die Programmlaufzeit über einen Watchdog überwacht.

Wird die maximale Watchdog-Zeit überschritten:
> Gerät geht in den Zustand "Timeout Error"
> alle Prozesse werden angehalten (Reset)
> alle Ausgänge werden ausgeschaltet
> der Bildschirm wird dunkel (BasicDisplay: CR0451)
> die Status-LED blinkt rot mit 10 Hz
Störung beseitigen:
- ▶ Applikation löschen!
- ▶ PowerOn-Reset
- ▶ Applikation neu ins Gerät laden

Je nach Hardware haben die einzelnen Geräte ein unterschiedliches Zeitverhalten:

Gerät	Watchdog [ms]
BasicController: CR040n	100
BasicDisplay: CR0451 (Applikations-Programm) BasicDisplay: CR0451 (Visualisierung)	100 1.200

Sollte der beschriebene Watchdog ausfallen, dann:
> ein zweiter Watchdog führt das Gerät in den Zustand "Fatal Error"
> die Status-LED leuchtet rot
Störung beseitigen:
- ▶ PowerOn-Reset

Wenn ohne Erfolg:
- ▶ Goto Bootloader
- ▶ PowerOn-Reset
- ▶ BasicSystem neu ins Gerät laden
- ▶ Applikation neu ins Gerät laden

Wenn ohne Erfolg:
- ▶ Hardware-Fehler: Gerät an ifm einsenden!

© ifm electronic gmbh | www.ifm.com

6.1.3 Verfügbarer Speicher

8002

Gilt nur für folgende Geräte:
- BasicController: CR0401

Physikalischer Speicher	Physikalisch vorhandener FLASH-Speicher (nichtflüchtiger, langsamer Speicher)	1 536 KiB
	Physikalisch vorhandener SRAM ¹) (flüchtiger, schneller Speicher)	208 KiB
	Physikalisch vorhandener EEPROM (nichtflüchtiger, langsamer Speicher)	---
	Physikalisch vorhandener FRAM ²) (nichtflüchtiger, schneller Speicher)	2 KiB
Nutzung des FLASH-Speichers	Speicher reserviert für den Code der IEC-Applikation	512 KiB
	Speicher für Daten außerhalb der IEC-Applikation, die vom Anwender beschrieben werden können, wie z.B. Files, Bitmaps, Fonts	---
	Speicher für Daten außerhalb der IEC-Applikation, die vom Anwender mit FBs wie FLASHREAD bearbeitet werden	64 KiByte
RAM	Speicher für die von der IEC-Applikation reservierten Daten im RAM	32 KiB
Remanenter Speicher	Speicher für in der IEC-Applikation als VAR_RETAIN deklarierten Daten	128 Byte
	Speicher für in der IEC-Applikation als RETAIN vereinbarten Merker	128 Byte
	Vom Anwender frei verfügbarer remanenter Speicher Der Zugriff erfolgt über die FBs FRAMREAD, FRAMWRITE	---
	Vom Anwender frei verfügbarer FRAM ²) Der Zugriff erfolgt über Adressoperator	---

¹) SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.
²) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

6.1.4 Verfügbarer Speicher

8136

Gilt nur für folgende Geräte:
- BasicController: CR0403

Physikalischer Speicher	Physikalisch vorhandener FLASH-Speicher (nichtflüchtiger, langsamer Speicher)	1 536 KiB
	Physikalisch vorhandener SRAM ¹) (flüchtiger, schneller Speicher)	592 KiB
	Physikalisch vorhandener EEPROM (nichtflüchtiger, langsamer Speicher)	---
	Physikalisch vorhandener FRAM ²) (nichtflüchtiger, schneller Speicher)	2 KiB
Nutzung des FLASH-Speichers	Speicher reserviert für den Code der IEC-Applikation	512 KiB
	Speicher für Daten außerhalb der IEC-Applikation, die vom Anwender beschrieben werden können, wie z.B. Files, Bitmaps, Fonts	---
	Speicher für Daten außerhalb der IEC-Applikation, die vom Anwender mit FBs wie FLASHREAD bearbeitet werden	64 KiByte
RAM	Speicher für die von der IEC-Applikation reservierten Daten im RAM	128 KiB
Remanenter Speicher	Speicher für in der IEC-Applikation als VAR_RETAIN deklarierten Daten	128 Byte
	Speicher für in der IEC-Applikation als RETAIN vereinbarten Merker	128 Byte
	Vom Anwender frei verfügbarer remanenter Speicher Der Zugriff erfolgt über die FBs FRAMREAD, FRAMWRITE	---
	Vom Anwender frei verfügbarer FRAM ²) Der Zugriff erfolgt über Adressoperator	---

¹) SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.
²) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

© ifm electronic gmbh | www.ifm.com

6.2 Programmierhinweise für CoDeSys-Projekte

Inhalt

FB, FUN, PRG in CoDeSys....44
Zykluszeit beachten!....45
Speichern....45
Applikations-Programm erstellen....45
ifm-Maintenance-Tool nutzen....47
Verteilen der Applikations-Software....47

7426

Hier erhalten Sie Tipps zum Programmieren des Geräts.

► Beachten Sie die Hinweise im CoDeSys-Programmierhandbuch
→ ecomat*mobile*-DVD "Software, tools and documentation".

6.2.1 FB, FUN, PRG in CoDeSys

8473

In CoDeSys unterscheiden wir folgende Typen von Bausteinen (POUs):

FB = function block = Funktionsblock

- Ein FB kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein FB darf in einem Projekt mehrmals aufgerufen werden.
- Für jeden Aufruf muss eine Instanz deklariert werden.
- Erlaubt: Im FB aufrufen von FB und FUN.

FUN = function = Funktion

- Eine Funktion kann mehrere Eingänge, aber nur einen Ausgang haben.
- Der Ausgang ist vom gleichen Datentyp wie die Funktion selbst.

PRG = program = Programm

- Ein PRG kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein PRG darf in einem Projekt nur einmal aufgerufen werden.
- Erlaubt: im PRG aufrufen von PRG, FB und FUN.

HINWEIS

Funktionsblöcke dürfen NICHT in Funktionen aufgerufen werden.
Sonst: Bei der Ausführung stürzt das Applikations-Programm ab.

Alle Bausteine (POUs) dürfen NICHT rekursiv aufgerufen werden, auch nicht indirekt.

Hintergrund:

Alle Variablen von Funktionen...
- werden beim Aufruf initialisiert und
- werden nach der Rückkehr zum Aufrufer ungültig.

Funktionsbausteine haben 2 Aufrufe:
- einen Initialisierungsaufruf und
- den eigentlichen Aufruf, um irgend etwas zu tun.

Folglich heißt das für den FB-Aufruf in einer Funktion, dass jedesmal ein zusätzlicher Initialisierungsaufruf über die Schnittstelle ginge.

© ifm electronic gmbh | www.ifm.com

6.2.2 Zykluszeit beachten!

8006

Bei den frei programmierbaren Geräten aus der Controller-Familie ecomatmobile stehen in einem großen Umfang Bausteine zur Verfügung, die den Einsatz der Geräte in den unterschiedlichsten Applikationen ermöglichen.

Da diese Bausteine je nach Komplexität mehr oder weniger Systemressourcen belegen, können nicht immer alle Bausteine gleichzeitig und mehrfach eingesetzt werden.

ACHTUNG

Gefahr von zu trägem Verhalten des Controllers! Zykluszeit darf nicht zu lang werden!

► Beim Erstellen des Applikations-Programms müssen die oben aufgeführten Empfehlungen beachtet und durch Austesten überprüft werden. Bei Bedarf muss durch Neustrukturieren der Software und des Systemaufbaus die Zykluszeit vermindert werden.

6.2.3 Speichern

7430

Gilt nur für folgende Geräte:
- Controller CRnn32, CRnn33
- PDM360: CR1050, CR1051
- PDM360compact: CR1052, CR1053, CR1055, CR1056
- PDM360NG: CR108n

Nur Dateien im Flash-Speicher (oder im EEPROM) sind vor Spannungsausfall geschützt.

Speichern Sie im Gerät zusammen mit Ihrem CoDeSys-Projekt immer auch das zugehörige Boot-Projekt:

► Menü [Online] > [Bootprojekt erzeugen] (dies muss nach jeder Änderung erneut erfolgen!).

> Nach einem Neustart startet das Gerät mit dem zuletzt gespeicherten Bootprojekt.

6.2.4 Applikations-Programm erstellen

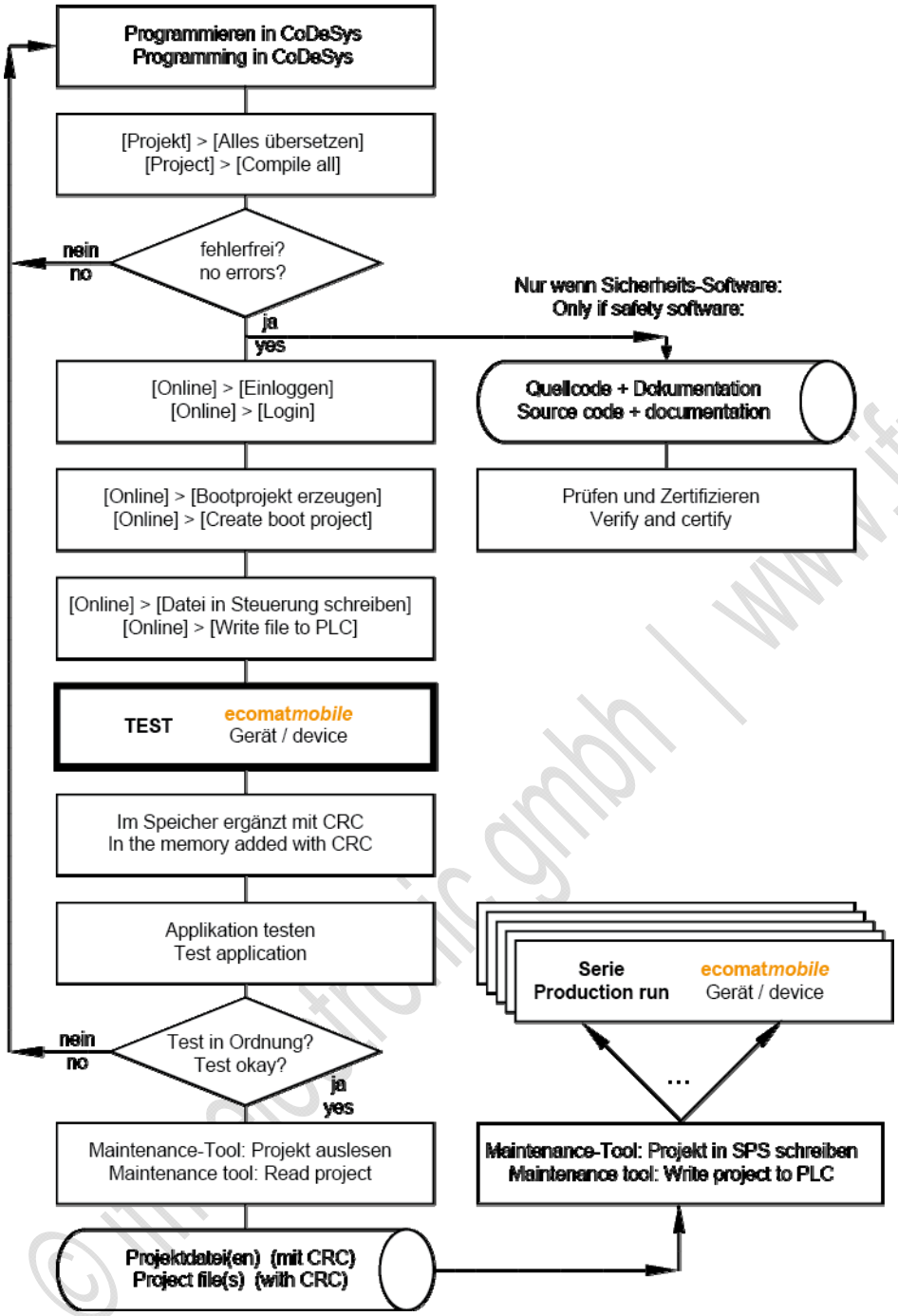
8490

Das Applikations-Programm wird mit dem Programmiersystem CoDeSys erstellt und während der Programmentwicklung mehrfach zum Testen in die Steuerung geladen:
In CoDeSys: [Online] > [Datei in Steuerung schreiben].

Für jeden derartigen Download via CoDeSys wird dazu der Quellcode neu übersetzt. Daraus resultiert, dass auch jedes Mal im Speicher der Steuerung eine neue Prüfsumme gebildet wird. Auch für Sicherheitssteuerungen ist dieses Verfahren bis zur Freigabe der Software zulässig.

© ifm electronic gmbh | www.ifm.com

Zumindest für sicherheitsrelevante Applikationen muss aber für die Serienproduktion der Maschine eine Einheitlichkeit der Software und ihrer Prüfsumme gewährleistet sein.



Grafik: Erstellen und Verteilen der (zertifizierten) Software

© ifm electronic gmbh | www.ifm.com

6.2.5 ifm-Maintenance-Tool nutzen

8492

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360: CR1050, CR1051
- PDM360compact: CR1052, CR1053, CR1055, CR1056
- PDM360NG: CR108n

Das ifm-Maintenance-Tool dient dem einfachen Übertragen des Programmcodes vom Programmierplatz in das Gerät. Grundsätzlich kann jede Applikations-Software mit dem ifm-Maintenance-Tool auf die Geräte kopiert werden. Vorteil: Dazu ist kein Programmiersystem mit einer CoDeSys-Lizenz erforderlich.

Sicherheitsrelevante Applikations-Software MUSS mit dem ifm-Maintenance-Tool auf die Steuerungen kopiert werden, um die Prüfsumme CRC, mit der die Software zertifiziert wurde, nicht zu verfälschen.

6.2.6 Verteilen der Applikations-Software

8493

Wir empfehlen folgenden Ablauf, wenn die Applikations-Software auf Serienmaschinen kopiert wird und zum Einsatz kommt:

- Sichern der Software
  Nach Abschluss der Programmentwicklung muss die letzte Version des in das Gerät geladenen Applikations-Programms mit dem ifm-Maintenance-Tool zunächst aus dem Gerät ausgelesen und auf einem Datenträger unter dem Namen `name_der_projektdatei.RESX` gespeichert werden. Nur dieses Verfahren gewährleistet, dass die Applikations-Software mit den entsprechenden Prüfsummen gesichert ist.
- Download der Software
  Um in der Serienproduktion alle Maschinen mit einer einheitlichen Software auszurüsten, darf nur diese Datei mit dem ifm-Maintenance-Tool in die Geräte geladen werden.
- Ein Fehler in den Daten dieser Datei wird durch die integrierte Prüfsumme beim erneuten Laden durch den ifm-Maintenance-Tool automatisch erkannt.

© ifm electronic gmbh | www.ifm.com

7 Fehlercodes und Diagnoseinformationen

Inhalt

Übersicht 48
Reaktion auf System-Fehler 48

1444

Um eine möglichst hohe Betriebssicherheit zu gewährleisten, wird das Gerät vom Betriebssystem in der Startphase (Reset-Phase) und während der Programmausführung durch interne Fehler-Checks überprüft.

7.1 Übersicht

8365

Folgende Systemmerker werden im Fehlerfall gesetzt:

> Die CAN-Funktionsblöcke liefern CAN-Fehlermeldungen an ihren Ausgängen RESULT.

> Der FB ***INPUT*** (→ Seite 212) liefert die Diagnosemeldungen der Eingänge an seinem Ausgang RESULT.

> Der FB ***OUTPUT*** (→ Seite 224) liefert die Diagnosemeldungen der Ausgänge an seinem Ausgang RESULT.

Vollständige Aufstellung der gerätespezifischen Fehlercodes und Diagnosemeldungen
→ Kapitel ***Systemmerker*** (→ Seite 264)

7.2 Reaktion auf System-Fehler

8504

Es liegt grundsätzlich in der Verantwortung des Programmierers, auf die Fehlermerker (Systemmerker) im Applikations-Programm zu reagieren.

Die spezifischen Fehlerbits und -bytes sollten im Applikations-Programm verarbeitet werden. Über den Fehlermerker erhält man eine Fehlerbeschreibung. Diese Fehlerbits/-bytes können bei Bedarf weiter verarbeitet werden.

> Das System setzt die Fehlermerker zurück, sobald der fehlerauslösende Zustand nicht mehr vorliegt.

7.2.1 Beispielablauf für Reaktion auf System-Fehler

8505

Das Betriebssystem schreibt zyklisch den Systemmerker TEMPERATURE.

Das Applikations-Programm erkennt die Gerätetemperatur durch Abfrage der INT-Variable.
Falls für die Applikation zulässige Werte über- oder unterschritten werden:

> Das Applikations-Programm schaltet die Ausgänge ab.

► Die Ursache des Fehlers beheben.

> Applikations-Programm erkennt den wieder normalen Temperaturwert:
Die Maschine / Anlage darf neu gestartet werden.

©ifm electronic gmbh | www.ifm.com

8 CAN einsetzen

Inhalt

Allgemeines zu CAN 49
Physikalische Anbindung des CAN 53
CAN-Datenaustausch 57
CAN-Bausteine auf Schicht 2 (RAW-CAN) 64
CAN-Bausteine nach SAE J1939 96
CAN-Bausteine nach CANopen 130
CAN-Fehler und Fehlerbehandlung 199

1163

8.1 Allgemeines zu CAN

Inhalt

Topologie 49
CAN-Schnittstellen 50
Verfügbare CAN-Schnittstellen und CAN-Protokolle 50
System-Konfiguration 52

1164

Der CAN-Bus (**C**ontroller **A**rea **N**etwork) gehört zu den Feldbussen.

Es handelt sich dabei um ein asynchrones, serielles Bussystem, das 1983 von Bosch für die Vernetzung von Steuergeräten in Automobilen entwickelt und 1985 zusammen mit Intel vorgestellt wurde, um die Kabelbäume (bis zu 2 km pro Fahrzeug) zu reduzieren und dadurch Gewicht zu sparen.

8.1.1 Topologie

1244

Das CAN-Netzwerk wird als Linienstruktur aufgebaut. Stichleitungen sind in eingeschränktem Umfang zulässig. Des Weiteren sind auch ein ringförmiger Bus (Infotainment Bereich) sowie ein sternförmiger Bus (Zentralverrieglung) möglich. Beide Varianten haben im Vergleich zum linienförmigen Bus jeweils einen Nachteil:

- Im ringförmigen Bus sind alle Steuergeräte in Reihe geschaltet, so dass bei einem Ausfall eines Steuergeräts der gesamte Bus ausfällt.
- Der sternförmige Bus wird meist von einem Zentralrechner gesteuert, da diesen alle Informationen passieren müssen, mit der Folge, dass bei einem Ausfall des Zentralrechners keine Informationen weitergeleitet werden können. Bei einem Ausfall eines einzelnen Steuergeräts funktioniert der Bus weiter.

Der lineare Bus hat den Vorteil, dass alle Steuergeräte parallel zu einer zentralen Leitung gehen. Nur wenn diese ausfällt, funktioniert der Bus nicht mehr.

HINWEIS

► Die Linie muss an ihren beiden Enden jeweils mit einem Abschlusswiderstand von der Größe 120 Ohm abgeschlossen werden, um ein Verfälschen der Signalqualität zu verhindern.

Die Geräte der **ifm electronic gmbh**, die mit einem CAN-Interface ausgestattet sind, haben grundsätzlich keine Abschlusswiderstände.

Stichleitungen und sternförmiger Bus haben den Nachteil, dass der Wellenwiderstand schwer zu bestimmen ist. Im schlimmsten Fall funktioniert der Bus nicht mehr.

©ifm electronic gmbh · www.ifm.com

8.1.2 CAN-Schnittstellen

7401

Je nach Aufbau der Hardware wird das Gerät mit mehreren CAN-Schnittstellen ausgerüstet. Grundsätzlich können alle Schnittstellen unabhängig voneinander mit folgenden Funktionen genutzt werden:

- Protokoll CAN auf Ebene 2 (→ Kapitel ***CAN-Bausteine auf Schicht 2 (RAW-CAN)*** (→ Seite 64))
- CANopen (→ Kapitel ***CAN-Bausteine nach CANopen*** (→ Seite 130)), ein Protokoll nach CiA 301/401 für Master/Slave-Betrieb (via CoDeSys)
- ***CANopen-Netzwerkvariablen*** (→ Seite 157) (via CoDeSys)
- Protokoll SAE J1939 (für Antriebs-Management, → Kapitel ***CAN-Bausteine nach SAE J1939*** (→ Seite 96))
- Buslast-Erkennung
- Errorframe-Zähler
- Download-Schnittstelle (nicht alle Geräte)
- 100 % Buslast ohne Paketverlust

Welche CAN-Schnittstelle des Geräts welche konkreten Möglichkeiten bietet, → Datenblatt des Geräts.
Das aktuelle Datenblatt finden Sie auf der ifm-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > Artikel-Nr.

Weitere interessante CAN-Protokolle sind:

- "Truck & Trailer Interface" nach ISO 11992
- ISOBUS nach ISO 11783 für Landmaschinen
- NMEA 2000 für den maritimen Einsatz
- CANopen Truck Gateway nach CiA 413 (Umsetzung zwischen ISO 11992 und SAE J1939)

8.1.3 Verfügbare CAN-Schnittstellen und CAN-Protokolle

6467

In den ecomatmobile-Geräten sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

Schnittstelle voreingestellter Download-ID **Gerät**	**CAN 1** ID 127	**CAN 2** ID 126	**CAN 3** ID 125	**CAN 4** ID 124	**Standard Baudrate** [kBit/s]
BasicController: CR040n	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	---	---	250
BasicDisplay: CR0451	CAN Layer 2 CANopen SAE J1939	---	---	---	250
CabinetController: CR0301, CR0302	CAN Layer 2 CANopen SAE J1939	---	---	---	125
CabinetController: CR0303	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 SAE J1939	---	---	125

© ifm electronic gmbh | www.ifm.com

Schnittstelle voreingestellter Download-ID **Gerät**	**CAN 1** ID 127	**CAN 2** ID 126	**CAN 3** ID 125	**CAN 4** ID 124	**Standard Baudrate** [kBit/s]
ClassicController: CR0020, CR0505	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 SAE J1939	---	---	125
ClassicController: CR0032, CR0033	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	125
ExtendedController: CR0200	**CPU 1 CAN 1** ID 127 CAN Layer 2 CANopen SAE J1939	**CPU 1 CAN 2** ID 126 CAN Layer 2 SAE J1939	**CPU 2 CAN 1** ID 127 CAN Layer 2 CANopen SAE J1939	**CPU 2 CAN 2** ID 126 CAN Layer 2 SAE J1939	125
ExtendedController: CR0232, CR0233	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	125
Platinensteuerung: CS0015	CAN Layer 2 CANopen SAE J1939	---	---	---	Drehschalter
SafetyController: CR7021, CR7506	CAN Layer 2 CANopen CANopen-Safety SAE J1939	CAN Layer 2 CANopen-Safety SAE J1939	---	---	125
ExtendedSafetyController: CR7201	**CPU 1 CAN 1** ID 127 CAN Layer 2 CANopen CANopen-Safety SAE J1939	**CPU 1 CAN 2** ID 126 CAN Layer 2 CANopen-Safety SAE J1939	**CPU 2 CAN 1** ID 127 CAN Layer 2 CANopen SAE J1939	**CPU 2 CAN 2** ID 126 CAN Layer 2 SAE J1939	125
SmartController: CR2500	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 SAE J1939	---	---	125
PDM360: CR1050, CR1051	CAN Layer 2 CANopen	CAN Layer 2 CANopen SAE J1939	---	---	125
PDM360compact: CR1052, CR1053, CR1055, CR1056	CAN Layer 2 CANopen	---	---	---	125
PDM360smart: CR1070, CR1071	CAN Layer 2 CANopen SAE J1939	---	---	---	125
PDM360NG: CR108n	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	CAN Layer 2 CANopen SAE J1939	125

© ifm electronic gmbh | www.ifm.com

8.1.4 System-Konfiguration

2270
8572

Die ecomatmobile-Geräte werden mit folgenden Download-Identifier (= ID) ausgeliefert:
- ID 127 für CAN-Schnittstelle 1
- ID 126 für CAN-Schnittstelle 2 (wenn vorhanden)
- ID 125 für CAN-Schnittstelle 3 (wenn vorhanden)
- ID 124 für CAN-Schnittstelle 4 (wenn vorhanden)

Das Download-System benutzt diesen Identifier für die erste Kommunikation mit einem nicht konfigurierten Modul über CAN.
Die Download-IDs können auf folgenden Wegen eingestellt werden:
- über den PLC-Browser des Programmiersystems,
- über den Downloader oder das MaintenanceTool oder
- über das Applikations-Programm.

Über den Modus "Autoconfig" des Bootloaders kann nur die CAN-Schnittstelle 1 eingestellt werden.

8571

Da der Download-Mechanismus auf Basis des CANopen-SDO-Dienstes arbeitet (auch wenn der CAN-Controller nicht im CANopen-Modus betrieben wird), müssen alle CAN-Controller im Netzwerk einen eindeutigen Identifier besitzen. Die eigentlichen COB-IDs werden nach dem "predefined connection set" aus den Modulnummern abgeleitet. Es darf jeweils nur ein nicht konfiguriertes Modul mit dem Netz verbunden werden. Nachdem die neue Teilnehmernummer 1...126 zugewiesen wurde, kann ein Download oder ein Debugging stattfinden und danach ein weiteres Gerät ins System eingebunden werden.

Der Download-ID wird unabhängig von dem CANopen-Identifier eingestellt. Es muss beachtet werden, dass sich diese IDs nicht mit den Download-IDs und den CANopen-Knotennummern der anderen CAN-Controller oder Netzwerkteilnehmer überschneiden.

Vergleich Download-ID vs. COB-ID:

Controller Programm-Download		CANopen	
Download-ID	**COB-ID SDO**	**Node-ID**	**COB-ID SDO**
1…127	TX: 580_{16} + Download-ID	1…127	TX: 580_{16} + Node-ID
	RX: 600_{16} + Download-ID		RX: 600_{16} + Node-ID

TX = Slave sendet an Master
RX = Slave empfängt von Master

HINWEIS

Der CAN-Download-ID des Geräts muss mit dem in CoDeSys eingestellten CAN-Download-ID übereinstimmen!

Im CAN-Netzwerk müssen die CAN-Download-IDs einmalig sein!

© ifm electronic gmbh | www.ifm.com

8.2 Physikalische Anbindung des CAN

Inhalt

Netzaufbau 53
CAN-Buspegel 54
Busleitungslänge 55
Leitungsquerschnitte 56

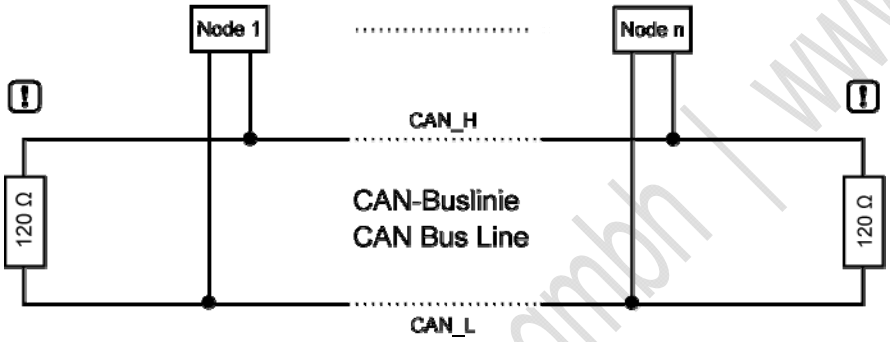
1177

Die in den Kapiteln ***CAN-Datenaustausch*** (→ Seite 57) und ***CAN-Fehler und Fehlerbehandlung*** (→ Seite 199) beschriebenen Mechanismen der Datenübertragung und der Fehlerbehandlung sind direkt im CAN-Controller implementiert. Die physikalische Verbindung der einzelnen CAN-Teilnehmer wird von der ISO 11898 in der Schicht 1 beschrieben.

8.2.1 Netzaufbau

1178

Die Norm ISO 11898 setzt einen Aufbau des CAN-Netzes mit einer Linienstruktur voraus.



Grafik: CAN-Netzaufbau Linienstruktur

HINWEIS

► Die Linie muss an ihren beiden Enden jeweils mit einem Abschlusswiderstand von der Größe 120 Ohm abgeschlossen werden, um ein Verfälschen der Signalqualität zu verhindern.

Die Geräte der **ifm electronic gmbh**, die mit einem CAN-Interface ausgestattet sind, haben grundsätzlich keine Abschlusswiderstände.

Stichleitungen

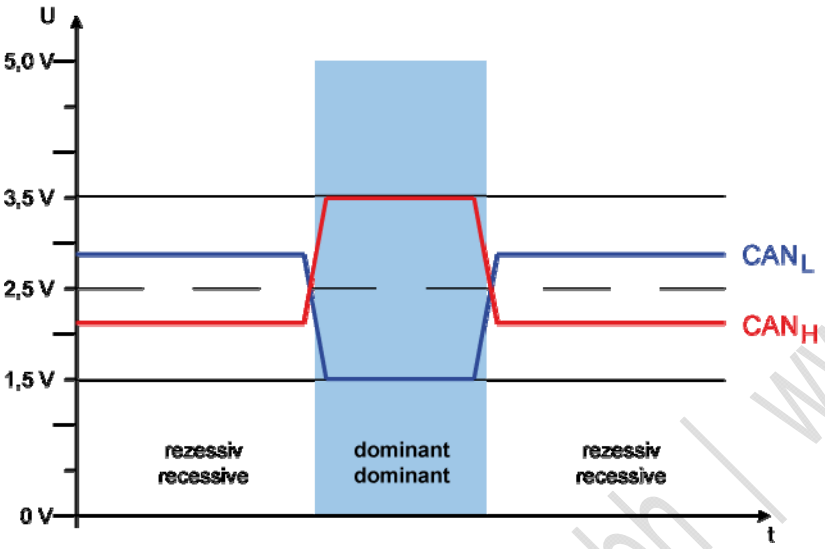
Idealerweise sollte zu den Busteilnehmern (Node 1 ... Node n) keine Stichleitung führen, da in Abhängigkeit von der Gesamtleitungslänge und den zeitlichen Abläufen auf dem Bus Reflektionen auftreten. Damit diese nicht zu Systemfehlern führen, sollten die Stichleitungen zu einem Busteilnehmer (z.B. einem E/A-Modul) eine gewisse Länge nicht überschreiten. Stichleitungen mit einer Länge von 2 m (bezogen auf 125 kBit/s) werden als unkritisch angesehen. Die Summe aller Stichleitungen im Gesamtsystem sollte 30 m nicht übersteigen. In besonderen Fällen müssen die Leitungslängen der Linie und der Stiche genau berechnet werden.

© ifm electronic gmbh | www.ifm.com

8.2.2 CAN-Buspegel

1179

Der CAN-Bus befindet sich im inaktiven (rezessiven) Zustand, wenn die Ausgangstransistorpaare in allen Busteilnehmern ausgeschaltet sind. Wird mindestens ein Transistorpaar eingeschaltet, wird ein Bit auf den Bus gegeben. Der Bus wird dadurch aktiv (dominant). Es fließt ein Strom durch die Abschlusswiderstände und erzeugt eine Differenzspannung zwischen den beiden Busleitungen. Die rezessiven und dominanten Zustände werden in den Busknoten in entsprechende Spannungen umgewandelt und von den Empfängerschaltkreisen erkannt.



Grafik: Buspegel

Durch diese differentielle Übertragung mit gemeinsamem Rückleiter wird die Übertragungssicherheit entscheidend verbessert. Störspannungen, die von außen auf das System einwirken, oder Massepotential-Verschiebungen beeinflussen beide Signalleitungen mit gleichen Störgrößen. Dadurch fallen die Störungen bei der Differenzbildung im Empfänger wieder heraus.

© ifm electronic gmbh | www.ifm.com

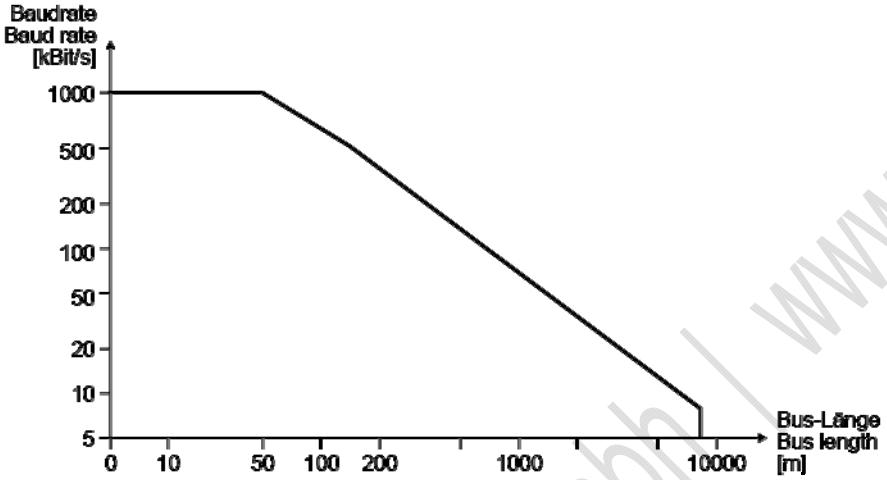
8.2.3 Busleitungslänge

1180

Die Länge der Busleitung ist abhängig von:

- Beschaffenheit der Busverbindung (Kabel, Steckverbinder),
- Leitungswiderstand,
- benötigte Übertragungsrate (Baud-Rate),
- Länge der Stichleitungen.

Vereinfachend kann man von folgender Abhängigkeit zwischen Buslänge und Baud-Rate ausgehen:



Grafik: Busleitungslänge

Baud-Rate [kBit/s]	Buslänge [m]	nominelle Bit-Länge [µs]
1 000	30	1
800	50	1,25
500	100	2
250	250	4
125	500	8
62,5	1 000	20
20	2 500	50
10	5 000	100

Tabelle: Abhängigkeiten Buslänge / Baudrate / Bitzeit

© ifm electronic gmbh | www.ifm.com

8.2.4 Leitungsquerschnitte

1181

Für die Auslegung des CAN-Netzes ist auch der Leitungsquerschnitt der eingesetzten Busleitung zu beachten. Die folgende Tabelle beschreibt die Abhängigkeit des Leiterquerschnitts bezogen auf die Leitungslänge und der Anzahl der daran angeschlossenen Teilnehmer (Knoten).

Leitungslänge [m]	**Leiterquerschnitt** [mm^2] **bei 32 Knoten**	**Leiterquerschnitt** [mm^2] **bei 64 Knoten**	**Leiterquerschnitt** [mm^2] **bei 100 Knoten**
≤ 100	0,25	0,25	0,25
≤ 250	0,34	0,50	0,50
≤ 500	0,75	0,75	1,00

Abhängig von den EMV-Anforderungen können Sie die Busleitungen wie folgt ausführen:
- parallel,
- als Twisted-Pair
- und/oder abgeschirmt.

© ifm electronic gmbh | www.ifm.com

8.3 CAN-Datenaustausch

Inhalt

Hinweise 58
CAN-Funktionen im IEC-User-Zyklus 60

7477

Der CAN-Datenaustausch erfolgt über das in der ISO 11898 international genormte CAN-Protokoll der Verbindungsschicht (Ebene 2) des siebenschichtigen ISO/OSI-Referenzmodells.

Jeder Bus-Teilnehmer kann Nachrichten senden (Multimaster-Fähigkeit). Der Datenaustausch arbeitet ähnlich dem Rundfunk. Daten werden ohne Absender und Adresse auf den Bus gesendet. Die Daten sind lediglich durch ihren Identifier gekennzeichnet. Es ist Aufgabe jedes Teilnehmers, die gesendeten Daten zu empfangen und an Hand des Identifiers zu prüfen, ob die Daten für diesen Teilnehmer relevant sind. Dieser Vorgang wird vom CAN-Controller in Verbindung mit dem Betriebssystem automatisch durchgeführt.

Je nach Gerät stehen folgende Baustein-Familien zur Verfügung:

- ***CAN-Bausteine auf Schicht 2 (RAW-CAN)*** (→ Seite 64):
  einfache Funktionen.
- ***CAN-Bausteine nach SAE J1939*** (→ Seite 96):
  hochwertige Funktionen für das Motor-Management.
- ***CAN-Bausteine nach CANopen*** (→ Seite 130):
  komplexe CAN-Funktionen.
- ***CANopen-Safety-Bausteine*** (optional):
  CAN-Funktionen für die Sicherheitsanwendungen in SafetyControllern.

Über diese Bausteine werden folgende Einheiten zu einem Datenobjekt verknüpft:

- die Arbeitsdaten,
- der Telegrammtyp,
- der gewählte Identifier (ID).

Diese Datenobjekte nehmen am Datenaustausch über den CAN-Bus teil. Die Sende- und Empfangsobjekte können aus allen gültigen IEC-Datentypen (z.B. BOOL, WORD, INT, ARRAY) definiert werden.

Die CAN-Nachricht besteht aus einem CAN-Identifier (***CAN-ID*** (→ Seite 58)) und maximal 8 Datenbytes. Der ID repräsentiert nicht das Absender- oder Empfängermodul, sondern kennzeichnet die Nachricht. Um Daten zu übertragen, ist es notwendig, dass im Sendemodul ein Sendeobjekt und in mindestens einem anderen Modul ein Empfangs-Objekt deklariert ist. Beide Deklarationen müssen dem gleichen Identifier und dem gleichen Telegrammtyp (base oder extended) zugeordnet sein.

© ifm electronic gmbh | www.ifm.com

8.3.1 Hinweise

8394

CAN-ID

1166

Je nach CAN-ID sind folgende CAN-Identifier frei verfügbar für den Datentransfer:

CAN-ID base	CAN-ID extended
11 Bit	29 Bit
2 047 CAN-Identifier	536 870 912 CAN-Identifier
Standard-Applikationen	Motor-Management (SAE J1939), Truck & Trailer Interface (ISO 11992)

HINWEIS

Der 29-Bit-CAN-ID steht bei einigen Geräten nicht für alle CAN-Schnittstellen zur Verfügung, → Datenblatt.

Derselbe CAN-Controller kann NICHT gleichzeitig 11 Bit und 29 Bit lange CAN-Identifier empfangen.

Wir empfehlen: In einem CAN-Netzwerk ausschließlich 11 Bit lange CAN-Identifier benutzen ODER 29 Bit lange CAN-Identifier.

Beispiel 11-Bit CAN-ID (base):

SOF	CAN-ID base Bit 28 ... Bit 18										RTR	IDE
0	0	0	0	0	1	1	1	1	1	1	0	0
	0			7				F				

Beispiel 29-Bit CAN-ID (extended):

SOF	CAN-ID base Bit 28 ... Bit 18										SRR	IDE	CAN-ID extended Bit 17 ... Bit 0																RTR
0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0		1			F				C			0			0			0			0							

Legende:
SOF = Start of frame
Flanke von rezessiv zu dominant

RTR = **R**emote **t**ransmission **r**equest
dominant: Diese Nachricht liefert Daten
rezessiv: Diese Nachricht fordert Daten an

IDE = **Id**entifier **e**xtension flag
dominant: Hiernach folgen Steuerungs-Bits
rezessiv: Hiernach folgt der zweite Teil des 29-Bit-Identifier

SRR = **S**ubstitute **r**emote **r**equest
rezessiv: Extended CAN-ID: Ersetzt das RTR-Bit an dieser Stelle

© ifm electronic gmbh | www.ifm.com

Zusammenfassung CAN / CANopen

7946

- Der COB-ID der Netzwerkvariablen muss sich unterscheiden vom CANopen-Slave-ID in der Steuerungskonfiguration und von den IDs der Sende- und Empfangs-Funktionsblöcke!
- Wenn mehr als 8 Bytes von Netzwerkvariablen in einen COB-ID gepackt werden, erweitert CANopen das Datenpaket automatisch auf mehrere aufeinander folgende COB-IDs. Dies kann zu Konflikten mit manuell definierten COB-IDs führen!
- Netzwerkvariablen können keine String-Variablen transportieren.
- Netzwerkvariablen können transportiert werden...
  - wenn eine Variable TRUE wird (Event),
  - bei Datenänderung in der Netzwerkvariablen oder
  - zyklisch nach Zeitablauf.
- Die Intervall-Zeit beschreibt die Periode zwischen Übertragungen bei zyklischer Übertragung. Der Mindestabstand beschreibt die Wartezeit zwischen zwei Übertragungen, wenn die Variable sich zu oft ändert.
- Um die Buslast zu mindern, die Nachrichten via Netzwerkvariablen oder Sende-FBs mit Hilfe von verschiedenen Events auf mehrere SPS-Zyklen verteilen.
- In der Steuerungskonfiguration sollten die Werte für [Com Cycle Period] und [Sync. Window Length] gleich groß sein.
- Wenn die [Com Cycle Period] für einen Slave eingestellt ist, sucht der Slave in genau dieser Zeit nach einem Sync-Objekt des Masters. Deshalb muss der Wert für [Com Cycle Period] größer sein als die [Master Synch Time].
- Wir empfehlen, Slaves als "optional startup" und das Netzwerk als "automatic startup" zu setzen. Dies reduziert unnötige Buslast und ermöglicht einem kurzzeitig verlorenen Slave, sich wieder in das Netzwerk zu integrieren.
- Wir empfehlen, Analog-Eingänge auf "synchrone Übertragung" zu setzen, um Busüberlastung zu vermeiden.
- Binäre Eingänge, insbesondere die unregelmäßig schaltenden, sollten am besten auf "asynchrone Übertragung" gesetzt werden. Um bei selten auftretenden Änderungen an den binären Eingängen trotzdem eine regelmäßige Übertragung des Status zu erreichen, sollte der Event-Timer aktiviert werden.
- Beim Überwachen des Slave-Status beachten:
  - Nach dem Starten von Slaves dauert es etwas, bis die Slaves "operational" sind.
  - Beim Abschalten des Systems können Slaves wegen vorzeitigem Spannungsverlust eine scheinbare Status-Änderung anzeigen.

© ifm electronic gmbh | www.ifm.com

8.3.2 CAN-Funktionen im IEC-User-Zyklus

Inhalt

Abarbeitung der Programmbausteine 60
Baustein-Eingänge 61
Baustein-Ausgänge 62
Bibliotheken für CAN-Bausteine 63

7396

IEC-User-Zyklus = SPS-Zyklus im CoDeSys-Applikations-Programm.

Die Bausteine der externen Bibliotheken werden NICHT während der Laufzeit des IEC-User-Zyklus, sondern zuvor oder danach ausgeführt. Dies bedeutet, dass die Funktionsblöcke oder Prozessvariablen aus Sicht des IEC-Programmierers während eines IEC-User-Zyklus statisch sind. Ausnahme sind Funktionsblöcke mit Sende-Funktionalität und FBs mit FiFo-Funktionalität:
- CAN_TX..., → ***FBs für RAW-CAN (Nachrichten senden)*** (→ Seite 73)
- ***CAN_RX_ENH_FIFO*** (→ Seite 88)
- ***CAN_RX_RANGE_FIFO*** (→ Seite 84)

FiFo (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Tatsächlich werden auch während des IEC-User-Zyklus die CAN-Stacks ausgeführt, also beispielsweise weiterhin Daten zyklisch gesendet oder MultiFrame-Nachrichten empfangen. Diese werden aber erst nach Ablauf des IEC-User-Zyklus dem Applikations-Programm zur Verfügung gestellt.

CAN-Stack = Stapel von CAN-Datenübertragungs-Aufträgen.

Abarbeitung der Programmbausteine

8352

Wichtig ist eine **geordnete Abarbeitung der Programmbausteine**. Das Betriebssystem stellt folgende Reihenfolge sicher:

- CAN_ENABLE
- J1939_ENABLE
- CANopen_ENABLE
- automatischer Start des CANopen-Netzwerkes aufgrund der entsprechenden Anwahl in der Entwicklungsumgebung
- Update Prozessabbild für J1939 und CANopen
- andere Funktionsblöcke

So wird z.B. beim Einstellen der Baudrate ein definierter Zustand auf einem System erreicht.

Die Baudrate, die bei CAN_ENABLE gewählt wurde, soll durch die nachfolgenden Aufrufe nicht mehr verändert werden.

Falls CAN_ENABLE auf FALSE gesetzt wird und damit der CAN-Kanal deaktiviert wurde, werden auch alle Protokolle zurückgesetzt.

©ifm electronic gmbh | www.ifm.com

Baustein-Eingänge

8353

Fast alle Bausteine weisen einen ENABLE- oder EXECUTE-Eingang auf:

- Der ENABLE-Eingang ist pegelbezogen. Solange er gesetzt ist oder bleibt, wird der Baustein ausgeführt und vor oder/und nach dem IEC-Task "gerechnet", also mit Daten befüllt.
- Ein EXECUTE-Eingang ist flankengesteuert. Der Baustein wird NUR beim Wechsel des Eingangs von 0 auf 1 einmalig ausgeführt.

Mit Eingang CHANNEL wird der CAN-Kanal gewählt.

Steht in den Listen bei einem Eingangs-Parameter der Zusatz "(Parameter-Nutzung optional)", dann kann der Applikations-Programmierer die Belegung dieses Eingangs weglassen. In diesem Fall arbeitet die Funktion mit der für diesen Eingang voreingestellten Belegung. Alle nicht so deklarierten Eingänge müssen belegt werden.

© ifm electronic gmbh | www.ifm.com

Baustein-Ausgänge

8354
7556

Einige Bausteine geben eine Ergebnis-Meldung RESULT zurück.

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
$1...31_{10}$		Globale Rückgabewerte; Beispiele:
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
004	04	FB ist in der Bearbeitung – Daten werden zyklisch bearbeitet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
006	06	FB ist in der Bearbeitung – Senden läuft noch
007	07	FB ist in der Bearbeitung – Remote für ID aktiv
008	08	Funktionsblock ist aktiv
014	0E	FB ist aktiv – CANopen-Master konfiguriert Slaves und sendet SDOs
015	0F	FB ist aktiv – CANopen-Master wird gestartet
$32_{10}...63_{10}$		FB-spezifische Rückgabewerte
$64_{10}…127_{10}$		FB-spezifische Fehlermeldungen
$128_{10}...255_{10}$		Globale Fehlermeldungen; Beispiele:
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Master konnte nicht gestartet werden
240	F0	Fehler: Mehrere modale Eingänge sind aktiv z.B. CANopen-NMT-Services
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich
247	F7	Fehler: Speicherüberschreitung (Länge größer als Array)
250	FA	Fehler: FiFo ist voll – Daten wurden verloren
252	FC	Fehler: CAN-Multiframe-Sendung ist fehlgeschlagen
253	FD	Fehler: CAN-Sendung ist fehlgeschlagen. Daten können nicht gesendet werden.
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

© ifm electronic gmbh | www.ifm.com

Bibliotheken für CAN-Bausteine

8355

Für CAN-Bausteine stellen wir verschiedene Bibliotheken zur Verfügung:

- ***CAN-Bausteine auf Schicht 2 (RAW-CAN)*** (→ Seite 64):
  Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`
- ***CAN-Bausteine nach SAE J1939*** (→ Seite 96):
  Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`
- ***CAN-Bausteine nach CANopen*** (→ Seite 130):
  Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

> Die Bibliotheken werden beim Anlegen des Projektes in CoDeSys NICHT automatisch geladen.

Der Applikations-Programmierer muss dafür sorgen, dass die empfangenen Daten rechtzeitig verarbeitet werden, bevor sie von nachfolgenden Daten überschrieben werden.

© ifm electronic gmbh | www.ifm.com

8.4 CAN-Bausteine auf Schicht 2 (RAW-CAN)

Inhalt

Hinweise 64
FBs für RAW-CAN (Verwaltung) 66
FBs für RAW-CAN (Nachrichten senden) 73
FBs für RAW-CAN (Nachrichten empfangen) 80
FBs für RAW-CAN (Remote-Zugriff) 91

7479

RAW-CAN bezeichnet das reine CAN-Protokoll, das ohne ein zusätzliches Kommunikationsprotokoll auf dem CAN-Bus (auf ISO/OSI-Schicht 2) arbeitet. Das CAN-Protokoll ist international nach ISO 11898-1 definiert und garantiert zusätzlich in ISO 16845 die Austauschbarkeit von CAN-Chips.

8.4.1 Hinweise

8468

FiFo-Begrenzungen

7555

FiFo (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Einige RAW-CAN-Bausteine ermöglichen das Versenden und Empfangen mehrerer Nachrichten innerhalb eines SPS-Zyklus, da die Nachrichten in einem FiFo zwischengespeichert werden:
- CAN_TX..., → ***FBs für RAW-CAN (Nachrichten senden)*** (→ Seite 73)
- ***CAN_RX_ENH_FIFO*** (→ Seite 88)
- ***CAN_RX_RANGE_FIFO*** (→ Seite 84)

Die Anzahl der FiFo-Nachrichten ist begrenzt. Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / Kriterium	BasicController: CR040n BasicDisplay: CR0451	PDM360NG: CR108n
max. FiFo senden - mit FB CAN_TX... - mit FB CAN_TX_ENH...	4 Nachrichten 16 Nachrichten	4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB CAN_RX_..._FIFO	32 Nachrichten	32 Nachrichten

© ifm electronic gmbh | www.ifm.com

FB-Rückmeldungen (Übersicht)

7556

Einige Bausteine geben eine Ergebnis-Meldung RESULT zurück.

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
$1...31_{10}$		Globale Rückgabewerte; Beispiele:
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
004	04	FB ist in der Bearbeitung – Daten werden zyklisch bearbeitet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
006	06	FB ist in der Bearbeitung – Senden läuft noch
007	07	FB ist in der Bearbeitung – Remote für ID aktiv
008	08	Funktionsblock ist aktiv
014	0E	FB ist aktiv – CANopen-Master konfiguriert Slaves und sendet SDOs
015	0F	FB ist aktiv – CANopen-Master wird gestartet
$32_{10}...63_{10}$		FB-spezifische Rückgabewerte
$64_{10}…127_{10}$		FB-spezifische Fehlermeldungen
$128_{10}...255_{10}$		Globale Fehlermeldungen; Beispiele:
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Master konnte nicht gestartet werden
240	F0	Fehler: Mehrere modale Eingänge sind aktiv z.B. CANopen-NMT-Services
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich
247	F7	Fehler: Speicherüberschreitung (Länge größer als Array)
250	FA	Fehler: FiFo ist voll – Daten wurden verloren
252	FC	Fehler: CAN-Multiframe-Sendung ist fehlgeschlagen
253	FD	Fehler: CAN-Sendung ist fehlgeschlagen. Daten können nicht gesendet werden.
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

© ifm Electronic GmbH | www.ifm.com

8.4.2 FBs für RAW-CAN (Verwaltung)

Inhalt

CAN_ENABLE 67
CAN_RECOVER 69
CAN_STATUS 70
CAN_SETDOWNLOADID 71

7557

Hier finden Sie Funktionsblöcke zur Verwaltung der CAN-Funktion auf der OSI-Schicht 2.

© ifm electronic gmbh | www.ifm.com

CAN_ENABLE

7492

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7494

Mit CAN_ENABLE wird die CAN-Hardware initialisiert. Ohne diesen Aufruf sind keine anderen Aufrufe im RAW-CAN möglich oder liefern einen Fehler zurück.

Zum Ändern der Baud-Rate ist folgender Ablauf erforderlich:

► Funktionsblock einen Zyklus lang auf ENABLE=FALSE halten.

> Alle Protokolle werden zurückgesetzt.

> Re-Initialisierung der CAN-Schnittstelle und auch der darauf laufenden CAN-Protokolle. Alle zum zyklischen Senden vorhanden Informationen gehen dabei ebenfalls verloren und müssen neu aufgesetzt werden.

> Beim erneuten ENABLE=TRUE wird die neue Baud-Rate übernommen.

Parameter der Eingänge

7495

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Schnittstelle wird initialisiert TRUE: Schnittstelle ist aktiv FALSE: diese Funktion wird nicht ausgeführt
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BAUDRATE	WORD	Baud-Rate [kBit/s] zulässige Werte = 20, 50, 100, 125, 250, 500, 800, 1 000 voreingestellt = 250

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8530

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	Funktionsblock ist aktiv
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CAN_RECOVER

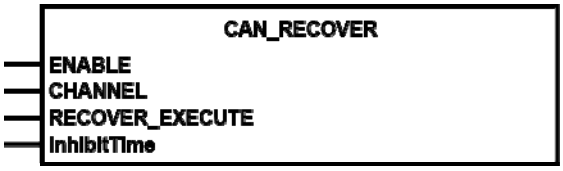
7512

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7513

CAN_RECOVER hat folgende Aufgaben:

- das automatische BusOff-Handling aktivieren / deaktivieren
- bei Vorliegen eines BusOff die CAN-Schnittstelle wieder neu starten.

> Wenn BusOff: CAN-Controller löscht alle Puffer (auch die Puffer der anderen Protokolle).

Wenn CAN_RECOVER nicht verwendet wird (ENABLE=FALSE):

> Bei einem BusOff wird automatisch nach 1 s ein Recover versucht.

> Nach 4 erfolglosen Recover-Versuchen in Folge wird die betroffene CAN-Schnittstelle deaktiviert.

Parameter der Eingänge

7514

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: KEIN automatisches Recover nach CAN-Busoff FALSE: Automatisches Recover nach CAN-Busoff
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RECOVER_EXECUTE	BOOL	TRUE (nur 1 Zyklus lang): Neustart der CAN-Schnittstelle Busoff-Zustand beheben FALSE: diese Funktion wird nicht ausgeführt
InhibitTime (Parameter-Nutzung optional)	TIME	Wartezeit zwischen BusOff und Neustart der CAN-Schnittstelle voreingestellt = 1 s

© ifm electronic gmbh | www.ifm.com

CAN_STATUS

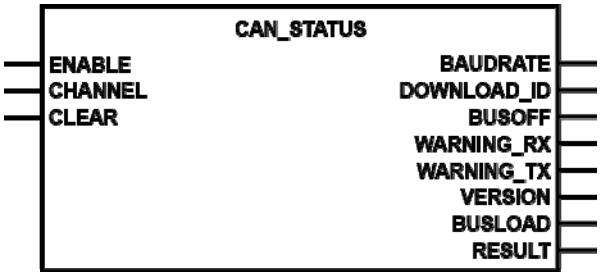
7499

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7501

Mit CAN_STATUS können Informationen zum gewählten CAN-Bus abgefragt werden.

Ohne Hardware-Initialisierung können folgende Merker wieder auf FALSE gesetzt werden:
- BUSOFF
- WARNING_RX
- WARNING_TX

Parameter der Eingänge

7502

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
CLEAR	BOOL	TRUE: Folgende Merker werden zurückgesetzt: - BUSOFF - WARNING_RX - WARNING_TX FALSE: diese Funktion wird nicht ausgeführt

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7504

Parameter	Datentyp	Beschreibung
BAUDRATE	WORD	aktuell eingestellte Baud-Rate
DOWNLOAD_ID	BYTE	aktueller Download-ID
BUSOFF	BOOL	Fehler CAN-BUSOFF an der Schnittstelle
WARNING_RX	BOOL	Empfangs-Warnschwelle an der Schnittstelle überschritten
WARNING_TX	BOOL	Sende-Warnschwelle an der Schnittstelle überschritten
VERSION	DWORD	Version des CAN-Stacks
BUSLOAD	BYTE	aktuelle Buslast in [%]
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist aktiv
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CAN_SETDOWNLOADID

7516

= Set Download-ID

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7517

Der Download-ID ist zum Datenaustausch erforderlich bei der Verbindung zwischen dem Laufzeitsystem und der CoDeSys-Entwicklungsumgebung. Der Download-ID wird voreingestellt beim Start des Geräts mit dem Wert aus der Hardware-Konfiguration gesetzt.

Mit CAN_DOWNLOADID kann dieser Wert im SPS-Programm (z.B. anhand bestimmter Eingänge) gesetzt werden. Der geänderte ID wird auch in die Hardware-Konfiguration geschrieben.

Parameter der Eingänge

7519

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DOWNLOAD_ID	BYTE	Download-ID (1...127) 0 = Download-ID lesen

Parameter der Ausgänge

7520

Parameter	Datentyp	Beschreibung
RESULT	BYTE	aktueller Download-ID

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	Funktionsblock ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.4.3 FBs für RAW-CAN (Nachrichten senden)

Inhalt

CAN_TX 73
CAN_TX_ENH....75
CAN_TX_ENH_CYCLIC 77

7610

Hier finden Sie Funktionsblöcke zum Senden von CAN-Nachrichten auf der OSI-Schicht 2.

© ifm electronic gmbh | www.ifm.com

CAN_TX

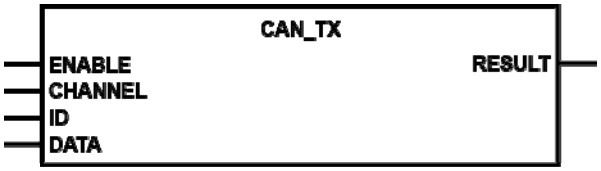
7522

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7523

CAN_TX ist die einfachste Sendefunktion. Mit ihr kann mit minimalem Wissen (und Ressourcen wie IEC-RAM) eine Standard-Nachricht pro Zyklus gesendet werden.

> Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Bei den einfachen Funktionen CAN_TX und CAN_RX wird anhand des ID ermittelt, ob ein Standard- oder ein Extended-Frame versendet werden soll. Bei den Enhanced-Versionen wird dies über den Eingang EXTENDED festgelegt. Mit den einfachen Funktionen kann man folglich keine Extended-Frames im ID-Bereich 0...2047 versenden.

Parameter der Eingänge

7524

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

CAN_TX_ENH

7558

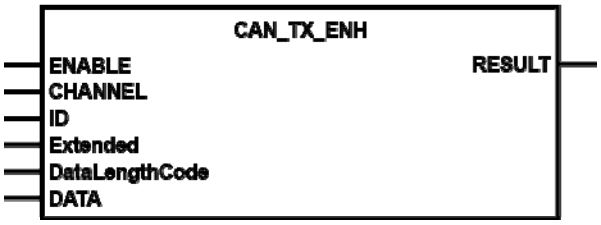
= CAN TX enhanced

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7559

Zusätzliche Einstellmöglichkeiten bietet CAN_TX_ENH (für: enhanced). Hier können alle CAN-spezifischen Eigenschaften individuell eingestellt werden, z.B.:

- Handelt es sich um einen 11- oder 29-Bit-Identifier?
- Die zusätzlichen Eingänge können voreingestellt werden, so dass ***CAN_TX*** (→ Seite 73) nicht erforderlich ist.

> Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)
DATALENGTHCODE	BYTE	Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

CAN_TX_ENH_CYCLIC

7568

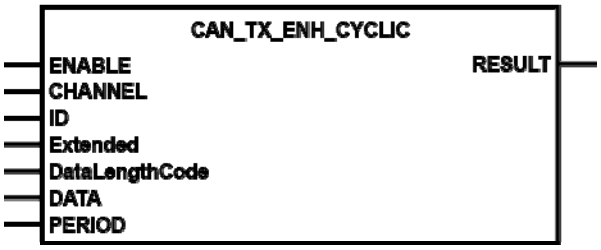
= CAN TX enhanced Cyclic

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7569

CAN_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.

Der FB entspricht ansonsten ***CAN_TX_ENH*** (→ Seite 75).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

Eine zu kurze Periodendauer kann zu einer hohen Buslast führen, was das Verhalten des Gesamtsystems beinträchtigen könnte.

Parameter der Eingänge

7582

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)
DataLengthCode (Parameter-Nutzung optional)	BYTE	Länge der zu sendenden Daten (0...8 Bytes) voreingestellt = 0 Bytes
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7510

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
008	08	Funktionsblock ist aktiv
009	09	CAN ist nicht aktiv
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

8.4.4 FBs für RAW-CAN (Nachrichten empfangen)

Inhalt

CAN_RX 80
CAN_RX_RANGE 82
CAN_RX_RANGE_FIFO 84
CAN_RX_ENH 86
CAN_RX_ENH_FIFO 88

7611

Hier finden Sie Funktionsblöcke zum Empfangen von CAN-Nachrichten auf der OSI-Schicht 2.

© ifm electronic gmbh | www.ifm.com

CAN_RX

7586

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7588

CAN_RX dient zum Empfang einer Nachricht. Die Eingabemöglichkeiten und der Speicherbedarf sind gering.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7590

Parameter	Datentyp	Beschreibung
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CAN_RX_RANGE

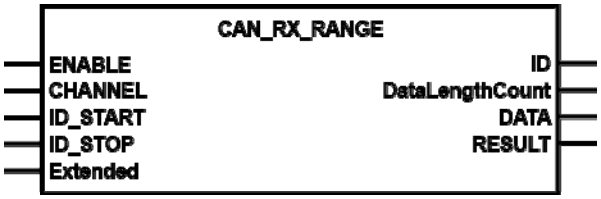
7592

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7594

CAN_RX_RANGE bietet folgende Einstellungen:

- den Nachrichten-Typ wählen (11 oder 29 Bit),
- einen Identifier-Bereich definieren.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7598

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})
DATALENGTHCOUNT	BYTE	Anzahl der empfangenen Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CAN_RX_RANGE_FIFO

7601

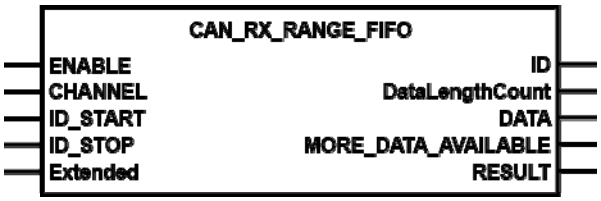
= CAN RX Range with FiFo

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7603

CAN_RX_RANGE_FIFO arbeitet grundsätzlich wie ***CAN_RX_RANGE*** (→ Seite 82).

Zusätzlich stellt CAN_RX_RANGE_FIFO ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

► Mit ENABLE die Funktion deaktivieren und wieder aktivieren.

> Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7604

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})
DATALENGTHCOUNT	BYTE	Anzahl der empfangenen Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten in der FiFo vorhanden FALSE: keine weiteren Daten in der FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

CAN_RX_ENH

7606

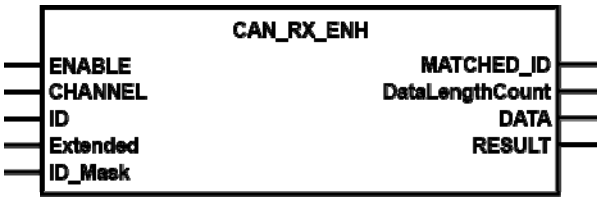
= CAN RX enhanced

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7608

CAN_RX_ENH bietet (im Vergleich zu ***CAN_RX*** (→ Seite 80)) zusätzlich folgende Möglichkeiten:

- den Frame-Typ wählen (11 oder 29 Bit),
- eine Maske für die Auswertung des CAN-ID definieren.

Bit-Vergleich von ID und Maske:	Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

Mit der Maske können Sie mehrere Identifier als Filter definieren. Beispiel:

ID =	100_{16} = 0001 0000 0000_2
ID_MASK =	$1F1_{16}$ = 0001 1111 0001_2
Ergebnis	Die CAN-IDs mit folgendem Bitmuster werden ausgewertet: xxx1 0000 $xxx0_2$ (x = beliebig), also für dieses Beispiel: 100_{16}, 102_{16}, 104_{16}, 106_{16}, 108_{16}, $10A_{16}$, $10C_{16}$, $10E_{16}$, 300_{16}, 302_{16}, 304_{16}, 306_{16}, 308_{16}, $30A_{16}$, $30C_{16}$, $30E_{16}$, 500_{16}, 502_{16}, 504_{16}, 506_{16}, 508_{16}, $50A_{16}$, $50C_{16}$, $50E_{16}$, 700_{16}, 702_{16}, 704_{16}, 706_{16}, 708_{16}, $70A_{16}$, $70C_{16}$, $70E_{16}$

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)
ID_Mask (Parameter-Nutzung optional)	DWORD	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein. voreingestellt = 0

Parameter der Ausgänge

7613

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})
DATALENGTHCOUNT	BYTE	Anzahl der empfangenen Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh - www.ifm.com

CAN_RX_ENH_FIFO

7615

= CAN RX enhanced with FiFo

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7616

CAN_RX_ENH_FIFO stellt (im Vergleich zu ***CAN_RX_ENH*** (→ Seite 86)) zusätzlich ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

► Mittels ENABLE den FB deaktivieren und wieder aktivieren.

> Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)
ID_Mask (Parameter-Nutzung optional)	DWORD	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein. voreingestellt = 0

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7617

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 2 048...536 870 912 (2^{29})
DATALENGTHCOUNT	BYTE	Anzahl der empfangenen Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten in der FiFo vorhanden FALSE: keine weiteren Daten in der FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

8.4.5 FBs für RAW-CAN (Remote-Zugriff)

Inhalt

CAN_REMOTE_REQUEST 91
CAN_REMOTE_RESPONSE 93

7623

Hier finden Sie Funktionsblöcke...

- zum Senden von CAN-Remote-Nachrichten
- zum Bereitstellen eines CAN-Telegramms, das beim Eintreffen einer Remote-Nachricht automatisch versendet werden soll.

© ifm electronic gmbh | www.ifm.com

CAN_REMOTE_REQUEST

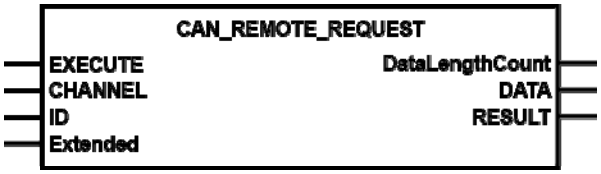
7625

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7627

Zur Anfrage einer Remote-Nachricht wird mit CAN_REMOTE_REQUEST eine entsprechende Anforderung versandt und die Antwort des anderen Gerätes als Ergebnis zurückgeliefert.

Parameter der Eingänge

7628

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7629

Parameter	Datentyp	Beschreibung
DATALENGTHCOUNT	BYTE	Anzahl der empfangenen Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist in der Bearbeitung – Empfangen läuft noch
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CAN_REMOTE_RESPONSE

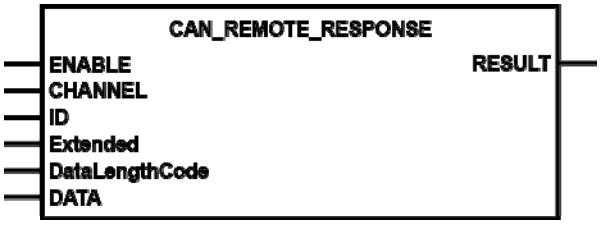
7631

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_RawCAN_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7633

CAN_REMOTE_RESPONSE stellt dem CAN-Controller im Gerät Daten zur Verfügung, die automatisch auf die Anfrage einer Remote-Nachricht gesendet werden.

Dieser FB ist stark geräte-abhängig. Es kann evtl. nur eine begrenzte Anzahl von Remote-Nachrichten eingerichtet werden:

BasicController: CR040n BasicDisplay: CR0451	max. 40 Remote-Nachrichten
PDM360NG: CR108n	max. 100 Remote-Nachrichten

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifier: Normal Frame: 0...2 047 (2^{11}) Extended Frame: 0...536 870 912 (2^{29})
Extended (Parameter-Nutzung optional)	BOOL	TRUE: Extended Frame FALSE: Normal Frame (voreingestellt für ID < 2 048)
DATALENGTHCODE	BYTE	Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7636

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
006	06	FB ist in der Bearbeitung – Remote für ID nicht aktiv
007	07	FB ist in der Bearbeitung – Remote für ID aktiv

© ifm electronic gmbh | www.ifm.com

8.5 CAN-Bausteine nach SAE J1939

Inhalt

CAN für die Antriebstechnik 97
Bausteine für J1939 (Verwaltung) 102
Bausteine für J1939 (Senden) 111
Bausteine für J1939 (Empfangen) 121

7482

Das Netzwerkprotokoll SAE J1939 beschreibt die Kommunikation auf einem CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B.Motordrehzahl, Temperatur) und Steuerungsinformationen.

© ifm electronic gmbh | www.ifm.com

8.5.1 CAN für die Antriebstechnik

Inhalt

Identifier nach SAE J1939 98
Beispiel: ausführliche Nachrichten-Dokumentation 99
Beispiel: kurze Nachrichten-Dokumentation 100
Leistungsgrenzen der Geräte (SAE J1939) 101

7678

Unter der Norm SAE J1939 bietet die CiA dem Anwender ein CAN-Busprotokoll für die Antriebstechnik an. Hierbei wird der CAN-Controller der Schnittstelle in den "Extended Mode" geschaltet. Das bedeutet, dass die CAN-Nachrichten mit einem 29 Bit-Identifier übertragen werden. Durch den längeren Identifier kann eine große Anzahl von Nachrichten direkt dem Identifier zugeordnet werden.

Bei der Protokollerstellung hat man sich diesen Vorteil zu Nutze gemacht und gruppiert bestimmte Nachrichten in ID-Gruppen. Die Zuordnung der IDs ist in den Normen SAE J1939 und ISO 11992 festgeschrieben.

Norm	Einsatzbereich
SAE J1939	Antriebs-Management
ISO 11992	"Truck & Trailer Interface"

Der 29 Bit-Identifier setzt sich aus zwei Teilen zusammen:
- einem 11 Bit-ID und
- einem 18 Bit-ID.

Vom Software-Protokoll unterscheiden sich die beiden Normen nicht, da die ISO 11992 auf der SAE J1939 aufbaut. Bezüglich der Hardwareschnittstelle besteht aber ein Unterschied: höhere Spannungspegel bei der ISO 11992.

Zur Nutzung der Funktionen nach SAE J1939 / ISO 11992 benötigt man auf jeden Fall die Protokollbeschreibung des Aggregat-Herstellers (z.B. für Motor, Getriebe). Aus dieser müssen die in das Aggregat-Steuergerät implementierten Nachrichten entnommen werden, da nicht jeder Hersteller alle Nachrichten implementiert oder die Implementierung nicht für alle Aggregate sinnvoll ist.

Folgende Informationen und Hilfsmittel sollten zur Entwicklung von Programmen für Funktionen nach SAE J1939 vorhanden sein:

- Aufstellung, welche Daten von den Aggregaten genutzt werden sollen
- Übersichtsliste des Aggregatherstellers mit allen relevanten Daten
- CAN-Monitor mit 29 Bit-Unterstützung
- Wenn benötigt, die Norm SAE J1939

© ifm electronic gmbh | www.ifm.com

Identifier nach SAE J1939

7675

Für den Datenaustausch unter SAE J1939 ist die Bildung des 29-Bit-Identifiers entscheidend. Dieser ist schematisch nachfolgend dargestellt:

A	SOF	Identifier 11 Bits											SRR	IDE	Identifier 18 Bits																	RTR
B	SOF	Priorität			R	DP	PDU Format (PF) 6+2 Bits						SRR	IDE	noch PF		PDU specific (PS) Ziel-Adresse Gruppe extern oder proprietär								Quell-Adresse							RTR
	1	3	2	1	1	1	8	7	6	5	4	3	1	1	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	1
C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
D	-	28	27	26	25	24	23	22	21	20	19	18	-	-	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	-

Legende:

A = CAN erweitertes Nachrichten-Format
B = J1939-Nachrichten-Format
C = J1939-Nachricht Bit-Position
D = CAN 29 Bit ID-Position

SOF = **S**tart **o**f **f**rame
SRR = **S**ubstitute **r**emote **r**equest
IDE = **Id**entifier **e**xtension flag
RTR = **R**emote **t**ransmission **r**equest
PDU = **P**rotocol **D**ata **U**nit
PGN = **P**arameter **G**roup **N**umber = PDU Format (PF) + PDU Source (PS)

(→ ***CAN-ID*** (→ Seite 58))

Dabei sind die 3 wesentlichen Kommunikationsmethoden unter SAE J1939 zu berücksichtigen:

- zielspezifische Kommunikation mit PDU1 (PDU-Format 0...239)
- Rundruf-Kommunikation mit PDU2 (PDU-Format 240...255)
- proprietäre Kommunikation mit PDU1 oder PDU2

© ifm electronic gmbh | www.ifm.com

Beispiel: ausführliche Nachrichten-Dokumentation

7679

ETC1: Electronic Transmission Controller #1 (3.3.5) $0CF00203_{16}$

Transmission repetition rate	RPT	10 ms
Data length	LEN	8 Bytes
PDU format	PF	240
PDU specific	PS	2
Default priority	PRIO	3
Data Page	PG	0
Source Address	SA	3
Parameter group number	PGN	$00F002_{16}$
Identifier	ID	$0CF00203_{16}$
Data Field	SRC	Die Bedeutung der Datenbytes 1...8 wird an dieser Stelle nicht weiter behandelt. Sie ist der Herstellerdokumentation zu entnehmen.

Da im Beispiel vom Hersteller alle relevanten Daten bereits aufbereitet wurden, können diese direkt an die Funktionsblöcke übertragen werden.

Dabei bedeuten:

Bezeichung in der Herstellerdokumentation	Baustein-Eingang Bibliotheksfunktion	Beispielwert
Transmission repetition rate	RPT	T#10ms
Data length	LEN	8
PDU format	PF	240
PDU specific	PS	2
Default priority	PRIO	3
Data Page	PG	0
Source Address / Destination Address	SA / DA	3
Data Field	SRC / DST	Array-Adresse

Je nach benötigter Funktion werden die entsprechenden Werte eingesetzt. Bei den Feldern SA / DA oder SRC / DST ändert sich die Bedeutung (aber nicht der Wert), entsprechend der Empfangs- oder der Sendefunktion.

Die einzelnen Datenbytes müssen aus dem Array ausgelesen und entsprechend ihrer Bedeutung weiterverarbeitet werden.

© ifm electronic gmbh | www.ifm.com

Beispiel: kurze Nachrichten-Dokumentation

7680

Aber auch wenn vom Aggregathersteller nur eine Kurzdokumentation zur Verfügung steht, kann man sich die FB-Parameter aus dem Identifier herleiten. Neben dem ID werden zusätzlich in jedem Fall die "Transmission repetition rate" und die Bedeutung der Datenfelder benötigt.

Wenn es sich nicht um herstellerspezifische Protokollnachrichten handelt, kann auch die Norm SAE J1939 oder ISO 11992 als Informationsquelle dienen.

Der Identifier $0CF00203_{16}$ setzt sich wie folgt zusammen:

PRIO, reserv., PG		PF + PS				SA / DA	
0	C	F	0	0	2	0	3

Da es sich bei diesen Werten um hexadezimale Zahlen handelt, von denen man teilweise einzelne Bits benötigt, müssen die Zahlen weiter zerlegt werden:

SA / DA		Source / Destination Address (hexadezimal)		Source / Destination Address (dezimal)	
0	3	00	03	0	3

PF		PDU format (PF) (hexadezimal)		PDU format (PF) (dezimal)	
F	0	0F	00	16	0

PS		PDU specific (PS) (hexadezimal)		PDU specific (PS) (dezimal)	
0	2	00	02	0	2

PRIO, reserv., PG		PRIO, reserv., PG (binär)	
0	C	0000	1100

Von den 8 Bit ($0C_{16}$) werden nur die 5 niederwertigen Bits benötigt:

nicht benötigt			Priority			res.	PG
x	x	x	0_2	1_2	1_2	0_2	0_2
x			03_{10}			0_{10}	0_{10}

Weitere typische Kombinationen für "PRIO, reserv., PG "

18_{16}:

nicht benötigt			Priority			res.	PG
x	x	x	1_2	1_2	0_2	0_2	0_2
x			6_{10}			0_{10}	0_{10}

$1C_{16}$:

nicht benötigt			Priority			res.	PG
x	x	x	1_2	1_2	1_2	0_2	0_2
x			7_{10}			0_{10}	0_{10}

© ifm electronic gmbh | www.ifm.com

Leistungsgrenzen der Geräte (SAE J1939)

7637

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / Kriterium	BasicController: CR040n BasicDisplay: CR0451	PDM360NG: CR108n
max. FiFo senden - mit FB J1939_TX - mit FB J1939_TX_ENH	 4 Nachrichten 16 Nachrichten	 4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB J1939_RX_FIFO	 32 Nachrichten	 32 Nachrichten
max. DTCs	64 Meldungen	64 Meldungen
max. Daten J1939	1 785 Bytes	1 785 Bytes

© ifm electronic gmbh | www.ifm.com

8.5.2 Bausteine für J1939 (Verwaltung)

Inhalt

J1939_ENABLE 103
J1939_NAME 104
J1939_GETDABYNAME 106
J1939_STATUS 108

7639

Hier finden Sie Funktionsblöcke zur Verwaltung der CAN-Funktion für SAE J1939.

© ifm electronic gmbh | www.ifm.com

J1939_ENABLE

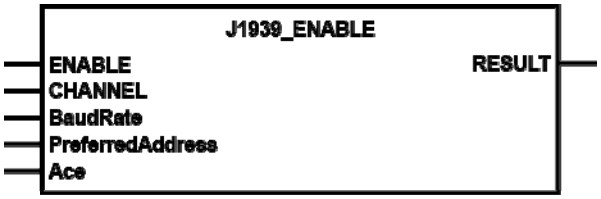
7641

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7642

Zur Initialisierung des J1939-Stack wird J1939_ENABLE auf TRUE=1 gesetzt.

> Dieser FB startet auch die Soft-I/Os aus der CFG-Datei.

> Eine andere Baud-Rate wird nur übernommen, wenn CAN_ENABLE nicht bereits aufgerufen wurde.

ACE = **A**ddress **C**laiming **E**nable = Freigabe Adressanforderung:

- Wenn ein ifm-Controller via J1939 mit nur einem Motorsteuergerät kommuniziert:
  dann ACE = FALSE setzen.
- Wenn jedoch mehrere Motorsteuergeräte am selben Bus arbeiten:
  dann ACE = TRUE setzen.
  In diesem Fall müssen die Motorsteuergeräte das Address Claiming auch unterstützen!
  Andernfalls riskieren Sie Adress-Überschneidungen mit nachfolgendem Systemausfall.

Parameter der Eingänge

7643

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): J1939-Stack wird initialisiert TRUE: J1939-Stack ist aktiv FALSE: J1939-Stack ist nicht in Betrieb
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BaudRate (Parameter-Nutzung optional)	WORD	Baud-Rate [kBit/s] zulässige Werte: 20, 50, 100, 125, 250, 500, 800, 1 000 Voreinstellung aus CFG-Datei, sonst 250 kBit/s
PreferredAddress (Parameter-Nutzung optional)	BYTE	Bevorzugte Quell-Adresse Voreinstellung aus CFG-Datei, sonst 252
Ace (Parameter-Nutzung optional)	BOOL	ACE = **A**ddress **C**laiming **E**nable = Freigabe Adressanforderung TRUE: Adressanforderung freigegeben (voreingestellt) (Steuergerät ist selbst-konfigurierend) FALSE: Keine Adressanforderung

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8542

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist aktiv
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_NAME

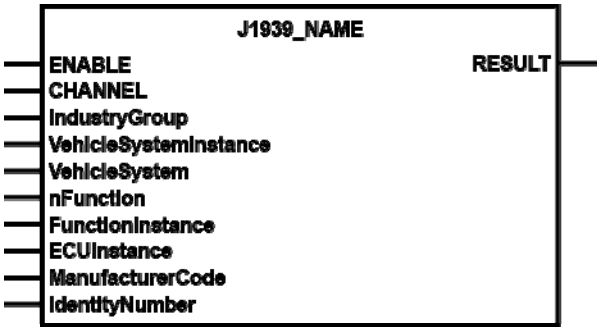
7646

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7648

Mit J1939_NAME kann dem Gerät ein Name gegeben werden, mit dem es sich im Netzwerk identifiziert.

Voreingestellt wird der Name der ifm verwendet.
Der Anwender hat die folgenden Möglichkeiten, den Namen des Gerätes zu ändern:

► die Informationen aus der CFG-Datei verwenden oder

► die gewünschten Daten mittels J1939_NAME überschreiben.

Die folgende Aufstellung zeigt die Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81:

Parameter	Datentyp	Beschreibung
Arbitrary Address Capable	1 Bit	beliebige Adresse verfügbar
Industry Group	3 Bit	Industriegruppe des Geräts
Vehicle System Instance	4 Bit	Instanz des Fahrzeugsystems
Vehicle System	7 Bit	Fahrzeugsystem
reserved	1 Bit	reserviert
Function	8 Bit	Funktion des Geräts
Function Instance	5 Bit	Instanz der Funktion
ECU Instance	3 Bit	Instanz der Steuerung
Manufacturer Code	11 Bit	Hersteller-Code (muss bei SAE beantragt werden)
Identify Number	21 Bit	Seriennummer des Geräts (sollte nicht überschrieben werden)

Tabelle: Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7652

Parameter	Datentyp	Beschreibung
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ENABLE	BOOL	TRUE: beliebige Adresse verfügbar FALSE: feste Adresse
IndustryGroup (Parameter-Nutzung optional)	BYTE	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 voreingestellt = 255_{10} = FF_{16} **)
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE	Instanz des Fahrzeugsystems zulässige Werte = 0...15 voreingestellt = 255_{10} = FF_{16} **)
VehicleSystem (Parameter-Nutzung optional)	BYTE	Fahrzeugsystem zulässige Werte = 0...127 voreingestellt = 255_{10} = FF_{16} **)
nFunction (Parameter-Nutzung optional)	WORD	Funktion des Geräts zulässige Werte = 0...255 voreingestellt = $65\,535_{10}$ = $FFFF_{16}$ **)
FunctionInstance (Parameter-Nutzung optional)	BYTE	Instanz der Funktion zulässige Werte = 0...31 voreingestellt = 255_{10} = FF_{16} **)
ECUInstance (Parameter-Nutzung optional)	BYTE	Instanz des Steuergeräts zulässige Werte = 0...7 voreingestellt = 255_{10} = FF_{16} **)
ManufacturerCode (Parameter-Nutzung optional)	WORD	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 (2^{11}-1) voreingestellt = $65\,535_{10}$ = $FFFF_{16}$ **)
IdentityNumber (Parameter-Nutzung optional)	DWORD	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 (2^{11}-1) voreingestellt = $4\,294\,967\,295_{10}$ = FFFF $FFFF_{16}$ **)

**) Der voreingestellte Wert wird nicht überschrieben.

Parameter der Ausgänge

7661

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_GETDABYNAME

7664

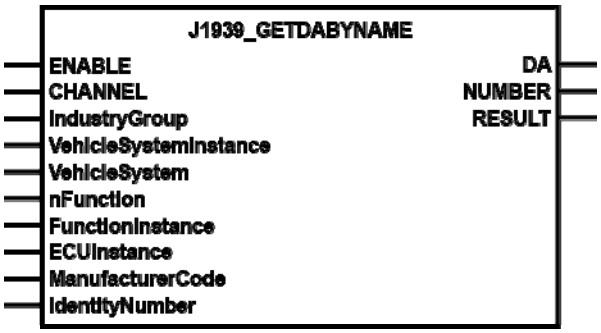
= Get Destination Arbitrary Name

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7665

Über J1939_GETDABYNAME lässt sich anhand der Namensinformation die Ziel-Adresse eines oder mehrerer anderer Teilnehmer bestimmen.

Wird an den optionalen Eingängen ein bestimmter Wert angelegt, werden in der Ergebnisliste nur die Teilnehmer aufgelistet, die diesen Wert besitzen. Wird kein oder der voreingestellte Wert eingestellt, wird bei der Filterung der Liste auf diesen Eintrag nicht geachtet.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7667

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IndustryGroup (Parameter-Nutzung optional)	BYTE	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 voreingestellt = 255_{10} = FF_{16} **)
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE	Instanz des Fahrzeugsystems zulässige Werte = 0...15 voreingestellt = 255_{10} = FF_{16} **)
VehicleSystem (Parameter-Nutzung optional)	BYTE	Fahrzeugsystem zulässige Werte = 0...127 voreingestellt = 255_{10} = FF_{16} **)
nFunction (Parameter-Nutzung optional)	WORD	Funktion des Geräts zulässige Werte = 0...255 voreingestellt = 65 535_{10} = $FFFF_{16}$ **)
FunctionInstance (Parameter-Nutzung optional)	BYTE	Instanz der Funktion zulässige Werte = 0...31 voreingestellt = 255_{10} = FF_{16} **)
ECUInstance (Parameter-Nutzung optional)	BYTE	Instanz des Steuergeräts zulässige Werte = 0...7 voreingestellt = 255_{10} = FF_{16} **)
ManufacturerCode (Parameter-Nutzung optional)	WORD	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 (2^{11}-1) voreingestellt = 65 535_{10} = $FFFF_{16}$ **)
IdentityNumber (Parameter-Nutzung optional)	DWORD	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 (2^{11}-1) voreingestellt = 4 294 967 295_{10} = FFFF $FFFF_{16}$ **)

**) Der voreingestellte Wert wird bei der Anfrage nicht berücksichtigt.

Parameter der Ausgänge

7668

Parameter	Datentyp	Beschreibung
DA	ARRAY (0...254) OF BYTE	Liste mit den gefundenen Teilnehmern. Einträge ohne gefundenen Teilnehmer werden mit 255 belegt.
NUMBER	BYTE	Anzahl der gefundenen Busteilnehmer
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	Funktionsblock ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_STATUS

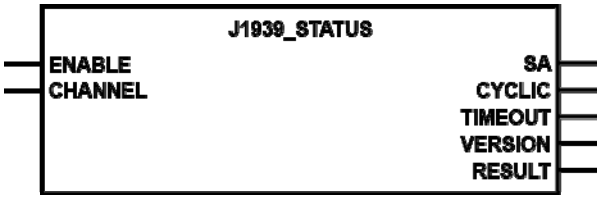
7670

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7672

Mit J1939_STATUS können relevante Informationen zum J1939-Stack zurückgelesen werden.

Parameter der Eingänge

7673

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7674

Parameter	Datentyp	Beschreibung
SA	BYTE	aktuelle Quell-Adresse (z.B. nach Adress-Claiming)
CYCLIC	WORD	Anzahl der zyklischen Nachrichten
TIMEOUT	BYTE	Quell-Adresse des Knotens, der Daten für Prozessabbild nicht rechtzeitig zur Verfügung gestellt hat 255_{10} = FF_{16} = alle Knoten haben rechtzeitig gesendet
VERSION	DWORD	Version des CAN-Stacks
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Protokoll aktiv
002	02	Protokoll inaktiv
003	03	Source Address claimed
004	04	Adresse verloren
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.5.3 Bausteine für J1939 (Senden)

Inhalt

J1939_TX 111
J1939_TX_ENH 112
J1939_TX_ENH_MULTI 114
J1939_TX_ENH_CYCLIC 116
J1939_DM1TX 118

7685

Hier finden Sie Funktionsblöcke zum Senden von CAN-Nachrichten über SAE J1939.

© ifm electronic gmbh | www.ifm.com

J1939_TX

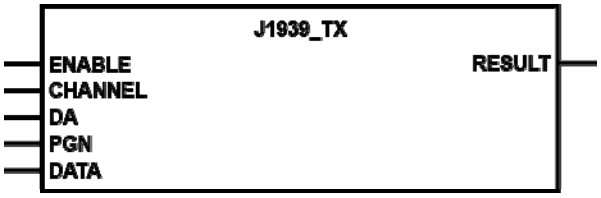
7688

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7689

J1939_TX ist die einfachste Methode zum Versenden von Single-Frame-Nachrichten.

Parameter der Eingänge

7690

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU Wenn PGN > 61139, dann wird Parameter ignoriert voreingestellt = 249_{10} = $F9_{16}$
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...$262\ 143_{10}$)
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7693

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

J1939_TX_ENH

7696

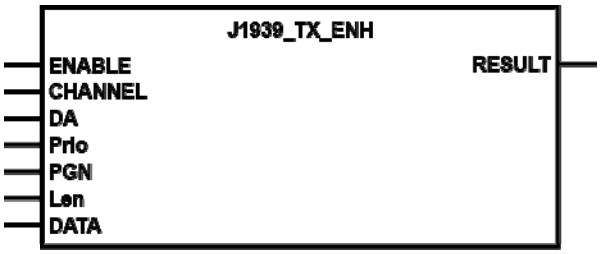
= J1939 TX enhanced

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7697

Zusätzliche Einstellmöglichkeiten bietet J1939_TX_ENH (für: enhanced) für Single-Frame-Nachrichten:

- Sende-Priorität
- Datenlänge

Multi-Frame Nachrichten → ***J1939_TX_ENH_MULTI*** (→ Seite 114).

Parameter der Eingänge

7702

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU Wenn PGN > 61139, dann wird Parameter ignoriert voreingestellt = 249_{10} = $F9_{16}$
Prio (Parameter-Nutzung optional)	BYTE	Nachrichten-Priorität zulässige Werte = 0…7 voreingestellt = 6
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...$262\ 143_{10}$)
Len (Parameter-Nutzung optional)	BYTE	Anzahl der zu sendenden Bytes zulässige Werte = 0...8 voreingestellt = 8
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7969

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

J1939_TX_ENH_MULTI

7699

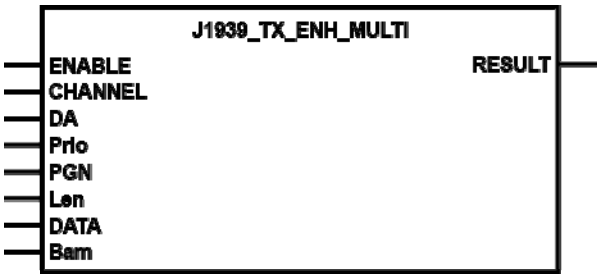
= J1939 TX enhanced Multiframe Message

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7705

Die Übertragung von Multi-Frame-Nachrichten erfolgt mit J1939_TX_ENH_MULTI.

Der FB entspricht ***J1939_TX_ENH*** (→ Seite 112). Zusätzlich kann hier bestimmt werden, ob die Übertragung als BAM (**B**roadcast **A**nnounce **M**essage) erfolgen soll.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7712

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU Wenn PGN > 61139, dann wird Parameter ignoriert voreingestellt = 249_{10} = $F9_{16}$
Prio (Parameter-Nutzung optional)	BYTE	Nachrichten-Priorität zulässige Werte = 0…7 voreingestellt = 6
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...262 143_{10})
Len (Parameter-Nutzung optional)	BYTE	Anzahl der zu sendenden Bytes zulässige Werte = 0...8 voreingestellt = 8
DATA	ARRAY [0...1784] OF BYTE	Zu sendende Daten (1...1785 Bytes)
Bam (Parameter-Nutzung optional)	BOOL	BAM = **B**roadcast **A**nnounce **M**essage = Nachricht an alle Teilnehmer TRUE: Multi-Frame-Übertragung als BAM Nachricht an alle Teilnehmer FALSE (voreingestellt): automatisch; Nachricht nur an Zieladresse

Parameter der Ausgänge

7714

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist aktiv
065	41	Senden ist nicht möglich
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_TX_ENH_CYCLIC

7716

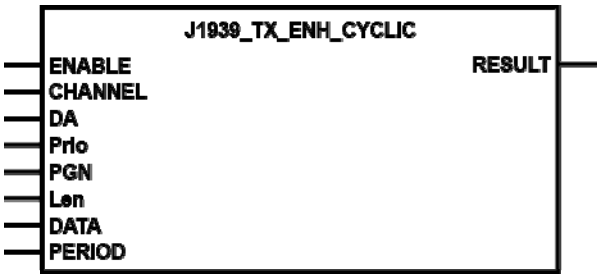
= J1939 TX enhanced Cyclic

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7718

J1939_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.

Der FB entspricht ansonsten ***J1939_TX_ENH*** (→ Seite 112).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

Eine zu kurze Periodendauer kann zu einer hohen Buslast führen
Die Buslast kann das Verhalten des Gesamtsystems beinträchtigen.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7719

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU Wenn PGN > 61139, dann wird Parameter ignoriert voreingestellt = 249_{10} = $F9_{16}$
Prio (Parameter-Nutzung optional)	BYTE	Nachrichten-Priorität zulässige Werte = 0…7 voreingestellt = 6
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...262 143_{10})
Len (Parameter-Nutzung optional)	BYTE	Anzahl der zu sendenden Bytes zulässige Werte = 0...8 voreingestellt = 8
DATA	ARRAY [0...7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

Parameter der Ausgänge

7720

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
008	08	Funktionsblock ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_DM1TX

7746

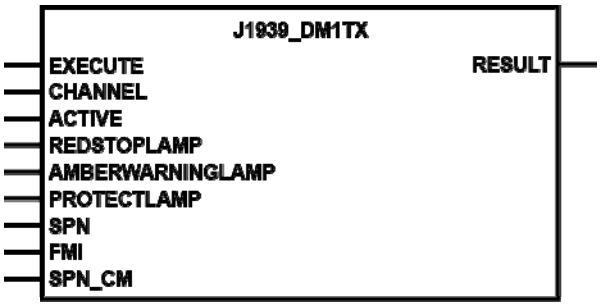
= J1939 Diagnostic Message 1 TX

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7747

Mit J1939_TX_DM1 (DM = **D**iagnostic-**M**essage) kann die Steuerung nur eine aktive Fehlermeldung an den CAN-Stack übergeben.

> Diese Meldung wird in die Hardware-Konfiguration gesichert

> Meldung wird als aktiv markiert und sekündlich als DM1 gesendet.

> Falls der Fehler bereits auftrat, wird der Ereignis-Zähler inkrementiert.

Der Ereignis-Zähler wird vom CAN-Stack verwaltet.

> Es erfolgt eine ODER-Verknüpfung aller Bits der Trouble-Codes. Sobald in einem der Trouble-Codes ein Bit gesetzt ist, ist es auch im Lampenstatus gesetzt.

Sobald eine Anfrage nach DM2 kommt, kann der CAN-Stack die entsprechenden Informationen aus der Hardware-Konfiguration auslesen und versenden.

> Bei Eintreffen einer DM3-Nachricht werden alle nicht aktiven Fehler im Fehlerspeicher in der Hardware-Konfiguration gelöscht.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7748

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ACTIVE	BOOL	TRUE: DTC ist aktiv Wird zyklisch (1x je Sekunde) als DM1 gesendet FALSE: DTC ist nicht mehr aktiv Wird in der Hardware-Konfiguration gesichert Wird bei Anfrage als DM2 gesendet
REDSTOPLAMP	BOOL	Red-Stop-Lamp TRUE: AUS FALSE: EIN
AMBERWARNINGLAMP	BOOL	Amber-Warning-Lamp TRUE: AUS FALSE: EIN
PROTECTLAMP	BOOL	ProtectLamp TRUE: AUS FALSE: EIN
SPN	WORD	**S**uspect **P**arameter **N**umber
FMI	BYTE	**F**ailure-**M**ode-**I**ndicator zulässige Werte = 00...$1F_{16}$ = 0...31_{10}
SPN_CM	BOOL	Conversion Method

Parameter der Ausgänge

7750

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Daten wurden in Fehlerspeicher aktiv gekennzeichnet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.5.4 Bausteine für J1939 (Empfangen)

Inhalt

J1939_RX 121
J1939_RX_FIFO 123
J1939_RX_MULTI 125
J1939_DM1RX 127

7722

Hier finden Sie Funktionsblöcke zum Empfangen von CAN-Nachrichten über SAE J1939.

© ifm electronic gmbh | www.ifm.com

J1939_RX

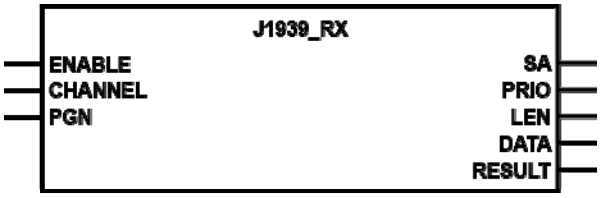
7724

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7725

J1939_RX ist die einfachste Methode zum Empfangen von Single-Frame-Nachrichten. Es wird die zuletzt auf dem CAN-Bus gelesene Nachricht zurückgegeben.

Parameter der Eingänge

7726

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...262 143_{10})

Die PGN = 0 wird nicht verwendet.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7727

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Priorität (0…7)
LEN	WORD	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
005	05	FB ist aktiv – noch keine Daten empfangen
009	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_RX_FIFO

7732

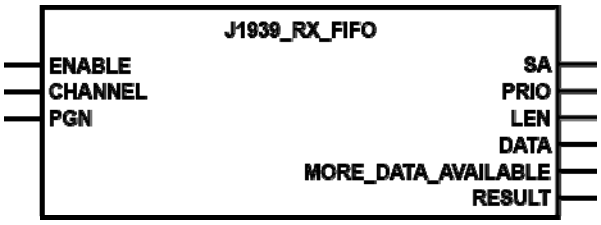
= J1939 RX with FiFo

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7733

J1939_RX_FIFO ermöglicht es, alle spezifizierten Nachrichten zu empfangen und nacheinander aus einem FiFo zu lesen.

Parameter der Eingänge

7734

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...262 143_{10})

Die PGN = 0 wird nicht verwendet.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7735

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Priorität (0…7)
LEN	BYTE	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0...7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten in der FiFo vorhanden FALSE: keine weiteren Daten in der FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
005	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh | www.ifm.com

J1939_RX_MULTI

7736

= J1939 RX Multiframe Message

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7741

Mit J1939_RX_MULTI ist der Empfang von Multiframe-Nachrichten möglich.

Parameter der Eingänge

7743

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässige Werte = 0000 0000...0003 $FFFF_{16}$ (0...262 143_{10})

Die PGN = 0 wird nicht verwendet.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7744

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Priorität (0…7)
LEN	WORD	Anzahl der empfangenen Bytes zulässige Werte = 0000 0000...0000 $06F9_{16}$ = 0...1 785_{10}
DATA	ARRAY [0...1784] OF BYTE	Zu sendende Daten (1...1785 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
005	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

J1939_DM1RX

7758

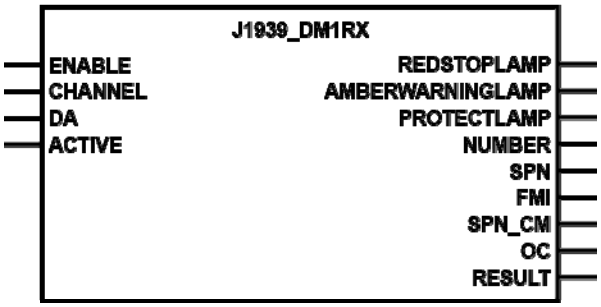
= J1939 Diagnostic Message 1 RX

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_J1939_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7761

J1939_RX_DM1 dient zum Empfang von Diagnosemeldungen DM1 oder DM2 von anderen ECUs.

Parameter der Eingänge

7762

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU, von der die DTCs geholt werden sollen. Bei 255 werden beliebige DM1- und DM2-Meldungen eingetragen, es können Nachrichten fehlen
ACTIVE	BOOL	TRUE: aktive DTCs (DM1) FALSE: davor aktive DTCs (DM2)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7763

Parameter	Datentyp	Beschreibung
REDSTOPLAMP	BOOL	Red-Stop-Lamp TRUE: AUS FALSE: EIN
AMBERWARNINGLAMP	BOOL	Amber-Warning-Lamp TRUE: AUS FALSE: EIN
PROTECTLAMP	BOOL	ProtectLamp TRUE: AUS FALSE: EIN
NUMBER	BYTE	Anzahl der empfangenen DTCs (0...8)
SPN	WORD	**S**uspect **P**arameter **N**umber
FMI	BYTE	**F**ailure-**M**ode-**I**ndicator zulässige Werte = 00...$1F_{16}$ = 0...31_{10}
SPN_CM	BOOL	Conversion Method
OC	BYTE	Occurrence Count = Ereigniszähler
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	FB ist aktiv – keine Daten wurden empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.6 CAN-Bausteine nach CANopen

Inhalt

Technisches zu CANopen 130
ifm-CANopen-SDO-Bausteine 164
CANopen-Netzwerk-Management 175

7485

HINWEIS

Folgende Geräte unterstützen CANopen nur für die 1. CAN-Schnittstelle:
- CabinetController: CR0303
- ClassicController: CR0020, CR0505
- ExtendedController: CR0200 (1. und 3. CAN-Schnittstelle)
- SafetyController: CR7021, CR7506
- ExtendedSafetyController: CR7201 (1. und 3. CAN-Schnittstelle)
- SmartController: CR25nn

Wurde bereits der CANopen-Master eingefügt, kann das Gerät nicht mehr als CANopen-Slave über CoDeSys genutzt werden.

Die Implementierung eines eigenen Protokolls auf Schnittstelle 2 oder Nutzung des Protokolls nach SAE J1939 oder ISO 11992 ist aber jederzeit möglich, sofern auf dem Gerät verfügbar (→ Kapitel ***Verfügbare CAN-Schnittstellen und CAN-Protokolle*** (→ Seite 50)).

Das Netzwerkprotokoll CANopen beschreibt ein standardisiertes Protokoll für verteilte industrielle Automatisierungssysteme auf der Basis von CAN. CANopen wurde im Rahmen der Nutzer- und Herstellervereinigung CiA entwickelt und ist seit Ende 2002 als EN 50325-4 standardisiert.

8.6.1 Technisches zu CANopen

Inhalt

Überblick 131
CANopen-Unterstützung durch CoDeSys 132
CANopen-Master 134
CANopen-Slave 150
CANopen-Netzwerkvariablen 157

7773

CANopen-Tabellen (→ Seite 265) zur Übersicht finden Sie im Anhang.

© ifm electronic gmbH | www.ifm.com

Überblick

7780

Master / Slave

7771

Die Steuerung läuft für einen CAN-Kanal entweder im Master- oder im Slave-Betrieb. Ein kombinierter Betrieb ist nicht vorgesehen. Die Funktionsblöcke erkennen aus den Konfigurationsdaten, ob die Steuerung als Master oder als Slave am CAN-Netzwerk läuft.

Die meisten der hier aufgeführten Funktionsblöcke unterscheiden bei deren Aufruf nicht, ob die SPS als Master oder als Slave läuft. Nur einige wenige Bausteine werden ausschließlich im Slave-Modus betrieben.

SYNC-Funktionalität

7772

Bezüglich der SYNC-Funktionalität als Slave sind die CAN-Stacks asynchron zur IEC-Applikation. Dies bedeutet, dass beim SYNC die letzten gültigen Daten des Prozessabbildes gesendet werden.

Extended CAN-IDs

7775

CANopen sieht keine Extended IDs vor.

Leistungsgrenzen der Geräte (CANopen)

7776

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / Kriterium	BasicController: CR040n BasicDisplay: CR0451	PDM360NG: CR108n
max. Guarding-Fehler	32 Meldungen	128 Meldungen
max. SDO-Daten	2 048 Bytes	2 048 Bytes

© ifm electronic gmbh | www.ifm.com

CANopen-Unterstützung durch CoDeSys

1857

Allgemeines zu CANopen mit CoDeSys

7777

CoDeSys ist eines der führenden Systeme für die Programmierung von Steuerungssystemen nach dem internationalem Standard IEC 61131. Um CoDeSys für den Anwender interessanter zu gestalten, wurden viele wichtige Funktionen in das Programmiersystem integriert, darunter auch ein Konfigurator für CANopen. Mit diesem CANopen-Konfigurator können Sie CANopen-Netzwerke (in einigen Punkten eingeschränkt) unter CoDeSys konfigurieren.

Wir verwenden bei diesen Geräten nicht mehr die CoDeSys-Bibliothek. Es muss zwar eine Bibliothek ins Projekt geladen werden, diese dient aber nur als Dummy.

Der CANopen-Stack befindet sich immer auf dem Gerät (im Laufzeitsystem). Für die Daten wird aber nur bei Bedarf Speicher reserviert.

CANopen Begriffe und Implementation

1858

Nach der CANopen-Spezifikation gibt es keine Master und Slaves in einem CAN-Netz. Stattdessen gibt es nach CANopen einen NMT-Master (NMT = Netzwerk-Management), einen Konfigurationsmaster usw., immer mit der Vorstellung, dass alle Teilnehmer eines CAN-Netzes gleichberechtigt sind.

Die Implementierung geht davon aus, dass ein CAN-Netz als Peripherie einer CoDeSys-programmierbaren Steuerung dient. Demzufolge wird eine **ecomat*mobile***-Steuerung oder ein PDM360-Display im CAN-Konfigurator von CoDeSys als CANopen-Master bezeichnet. Dieser Master ist NMT-Master und Konfigurationsmaster. Im Normalfall wird der Master dafür sorgen, dass das Netz in Betrieb genommen werden kann. Er übernimmt die Initiative, die einzelnen Nodes (= Netzwerk-Knoten) zu starten, die ihm per Konfiguration bekannt sind. Diese Nodes werden als Slaves bezeichnet.

Um den Master ebenfalls dem Status eines CANopen-Slaves näherzubringen, wurde ein Objektverzeichnis für den Master eingeführt. Auch kann der Master als SDO-Server (SDO = Service Data Object) auftreten und nicht nur in der Konfigurationsphase der Slaves als SDO-Client.

© ifm electronic gmbh | www.ifm.com

IDs (Adressen) in CANopen

3952

In CANopen werden diverse Arten von 'Adressen' (hier: IDs) unterschieden:

- COB-ID
  Der **C**ommunication-**Ob**ject-**Id**entifier adressiert die Nachricht (= das Kommunikationsobjekt) im Geräteverzeichnis. Ein Kommunikationsobjekt besteht aus einem oder mehreren CAN-Nachrichten mit bestimmten Aufgaben, z.B.:
  - PDO (**P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten),
  - SDO (**S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten),
  - Emergency (Nachrichten-Objekt mit Notfalldaten),
  - Time (Nachrichten-Objekt mit Zeitangaben) oder
  - Error Control (Nachrichten-Objekt mit Fehlermeldungen).
- CAN-ID
  Der **CAN-Id**entifier definiert netzwerkweit CAN-Nachrichten. Der CAN-ID ist Hauptbestandteil des Arbitration-Feldes eines CAN-Datenübertragungsblocks. Je niedriger der CAN-ID, desto höher die Priorität der Meldung.
- Download-ID
  Der Download-ID bezeichnet den Node-ID für Service-Kommunikation per SDO für den Programm-Download und das Debuggen.
- Node-ID
  Der **Node-Id**entifier ist ein eindeutiger Bezeichner für CANopen-Geräte (Devices) im CAN-Netzwerk. Der Node-ID ist auch Bestandteil einiger vordefinierter Verbindungssätze (→ ***Funktions-Code / Predefined Connectionset*** (→ Seite 268)).

Vergleich Download-ID vs. COB-ID:

Controller Programm-Download		CANopen	
Download-ID	**COB-ID SDO**	**Node-ID**	**COB-ID SDO**
1…127	TX: 580_{16} + Download-ID	1…127	TX: 580_{16} + Node-ID
	RX: 600_{16} + Download-ID		RX: 600_{16} + Node-ID

TX = Slave sendet an Master
RX = Slave empfängt von Master

© ifm electronic gmbh | www.ifm.com

CANopen-Master

Inhalt

Abgrenzung zu anderen CANopen-Bibliotheken 134
Ein CANopen-Projekt erstellen 135
CANopen-Slaves einfügen und konfigurieren 137
Der Master zur Laufzeit 142
Netzwerk starten 144
Netzwerkzustände 144

1859

Abgrenzung zu anderen CANopen-Bibliotheken

8015

Es wird nicht der CANopen-Stack von 3S*) verwendet, der Stack kann aber trotzdem über den in CoDeSys integrierten CANopen-Konfigurator konfiguriert werden. Der Stack wurde nach der Spezifikation der CiA DS301, V402 erstellt.

*) 3S = Smart Software Solutions, → CoDeSys

Für Sie als Anwender ergeben sich folgende Vorteile:

- Das komplette System beinhaltet den CANopen-Konfigurator und die Einbindung in das Entwicklungssystem.
- Die Ressourcen des Zielsystems werden geschont, da nicht die Ressourcen für eine Maximalkonfiguration vorgehalten werden.
- Automatisches Aktualisieren der Ein- und Ausgänge ohne zusätzliche Maßnahmen.

Folgende in CANopen definierten Funktionen werden zurzeit von der ifm-CANopen-Bibliothek unterstützt:

- **PDOs Senden:** Master sendet zu den Slaves (Slave = Knoten, Device)
  Senden ereignisgesteuert (d.h. bei Änderung), zeitgesteuert (RepeatTimer) oder als synchrone PDOs, d.h. immer wenn ein SYNC vom Master gesendet wurde. Auch eine externe SYNC-Quelle kann benutzt werden, um das Senden von synchronen PDOs zu initiieren.
- **PDOs Empfangen:** Master empfängt vom Slave
  Je nach Slave: ereignisgesteuert, abfragegesteuert, azyklisch und zyklisch.
- **PDO-Mapping**
  Zuordnung zwischen lokalem Objektverzeichnis und PDOs vom/zum CANopen-Slave (wenn vom Slave unterstützt).
- **SDO Senden und Empfangen**
  Automatische Konfiguration aller Slaves über SDOs beim Systemstart.
  Applikationsgesteuertes Senden und Empfangen von SDOs zu konfigurierten Slaves.
- **Synchronisation**
  Automatisches Senden von SYNC-Nachrichten durch den CANopen-Master.
- **Nodeguarding**
  Automatisches Senden von Guarding-Nachrichten und Überwachung der Lifetime für jeden entsprechend konfigurierten Slave.
  Wir empfehlen: Für aktuelle Geräte besser mit Heartbeat arbeiten, weil dann die Buslast niedriger ist.
- **Heartbeat**
  Automatisches Senden und Überwachen von Heartbeat-Nachrichten.
- **Emergency**
  Empfangen und Speichern von Emergency-Nachrichten von den konfigurierten Slaves.

© ifm electronic gmbh | www.ifm.com

- **Node-ID** und **Baudrate** in den Slaves setzen
  Durch Aufruf einer einfachen Funktion können Node-ID und Baudrate eines Slaves zur Laufzeit der Applikation gesetzt werden.

Folgende in CANopen definierten Funktionen werden vom 3S-CANopen-Konfigurator derzeit **nicht** unterstützt:

- dynamische Identifier-Zuordnung
- dynamische SDO-Verbindungen
- segmentierter SDO-Transfer
  Funktionalität kann realisiert werden über folgende Funktionsblöcke aus der ifm-CANopen-Bibliothek:
  - ***CANOPEN_SDOREAD*** (→ Seite 166)
  - ***CANOPEN_SDOREADMULTI*** (→ Seite 168)
  - ***CANOPEN_SDOWRITE*** (→ Seite 170)
  - ***CANOPEN_SDOWRITEMULTI*** (→ Seite 172)
- Alle oben nicht genannten Möglichkeiten des CANopen Protokolls.

Ein CANopen-Projekt erstellen

8513

Die Erstellung eines neuen Projektes mit einem CANopen-Master wird nachfolgend schrittweise beschrieben. Dabei gehen wir davon aus, dass Sie CoDeSys auf dem Rechner bereits fertig installiert haben und die Target- und EDS-Dateien ebenfalls richtig installiert oder kopiert wurden.

Eine weitergehende detaillierte Beschreibung zur Einstellung und Anwendung des Dialogs Steuerungs- und CANopen-Konfiguration → CoDeSys-Handbuch unter [Ressourcen] > [Steuerungskonfiguration] und in der Online-Hilfe.

► Nach der Neuanlage eines Projektes (→ Kapitel ***Target einrichten*** (→ Seite 17)) in der Steuerungskonfiguration über [Einfügen] > [Unterelement anhängen] den CANopen-Master an der gewünschten CAN-Schnittstelle einfügen.

> Die folgenden Bibliotheken und Software-Module werden automatisch eingebunden:

- die `STANDARD.LIB`, welche die in der IEC-61131 definierten Standardfunktionen für die Steuerung zu Verfügung stellt,
- die Systembibliotheken `SysLibSem.LIB` und `SysLibCallback.LIB`.

► Zusätzlich die folgenden Bibliotheken und Software-Module einbinden:

- die Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`
- die Gerätebibliothek für die jeweilige Hardware, z.B. `ifm_CR0403_Vxxyyzz.LIB`. Diese Bibliothek stellt alle gerätespezifischen Funktionen zur Verfügung.
- EDS-Dateien für alle Slaves, die am Netzwerk betrieben werden sollen. Die EDS-Dateien für alle ifm-CANopen-Slaves stellt die **ifm electronic gmbh** zur Verfügung (→ Kapitel ***Programmiersystem über Templates einrichten****).
  Für die EDS-Dateien von Fremd-Knoten ist der jeweilige Hersteller verantwortlich.

*) Eine Beschreibung zu den Templates und Demos finden Sie in der Anleitung "ifm ecomatmobile Templates und Demos" auf unserer Homepage:
DE → https://www.ifm.com/ifmde/web/ecomat-download-login.htm
UK → https://www.ifm.com/ifmuk/web/ecomatic-download.htm
FR → https://www.ifm.com/ifmfr/web/ecomatic-download.htm

© ifm electronic gmbh | www.ifm.com

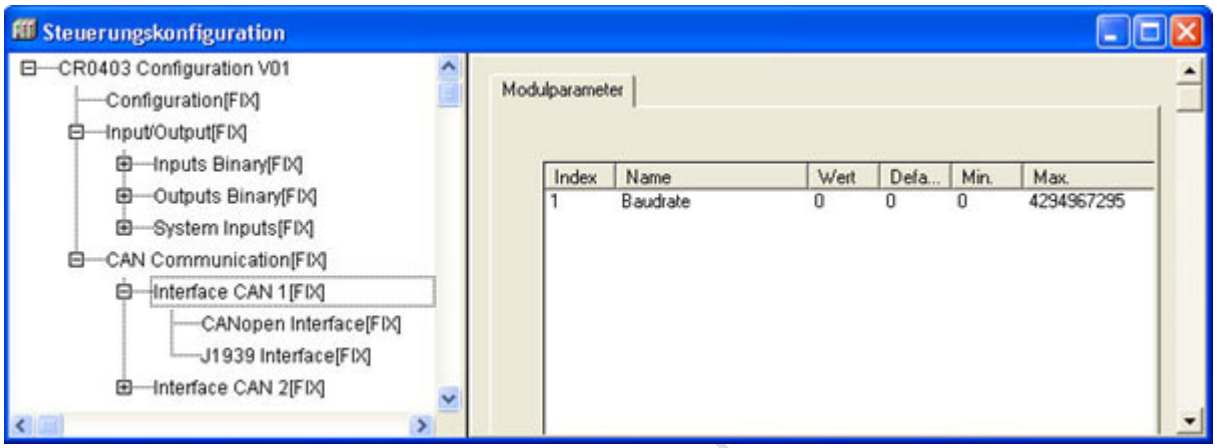
CANopen-Master: Register [CAN-Parameter]

7584

In diesem Dialogfenster können für den Master die wichtigsten Parameter eingestellt werden. Bei Bedarf kann über die Schaltfläche [EDS...] der Inhalt der Master-EDS-Datei angesehen werden.

Diese Schaltfläche wird nur angezeigt, wenn die EDS-Datei (z.B. `CR0020MasterODEntry.EDS`) im Verzeichnis `...\CoDeSys V2.3\Library\PLCConf` vorhanden ist.

Aus dieser EDS-Datei wird bei der Übersetzung des Applikations-Programms automatisch das Objektverzeichnis des Masters erzeugt.



© ifm electronic gmbh

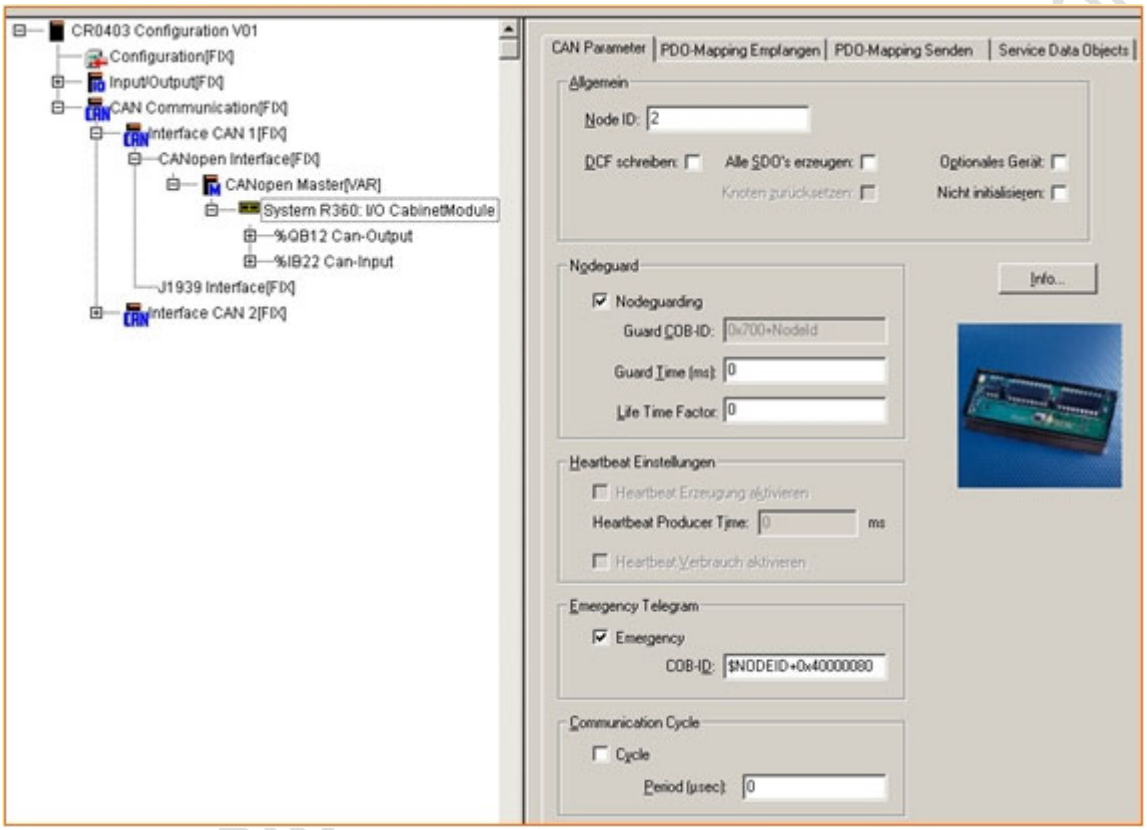
CANopen-Slaves einfügen und konfigurieren

Inhalt

CANopen-Slave: Register [CAN Parameter] 138
Register [PDO-Mapping empfangen] und [PDO-Mapping senden] 140
Register [Service Data Objects] 141

8516

Als nächstes können Sie nun die CANopen-Slaves einfügen. Dazu müssen Sie erneut den Dialog in der Steuerungskonfiguration [Einfügen] > [Unterelement anhängen] aufrufen. Es steht Ihnen eine Liste der im Verzeichnis PLC_CONF gespeicherten CANopen-Gerätebeschreibungen (EDS-Dateien) zur Verfügung. Durch Auswahl des entsprechenden Gerätes wird dieses direkt in den Baum der Steuerungskonfiguration eingefügt.



© ifm electronic

CANopen-Slave: Register [CAN Parameter]

1968

CAN Parameter: Node-ID

10036

Der Node-ID dient zur eindeutigen Identifizierung des CAN-Moduls und entspricht der am Modul eingestellten Nummer zwischen 1 und 127.

Der ID wird dezimal eingegeben und wird automatisch um eins erhöht, wenn Sie ein neues Modul hinzufügen.

CAN Parameter: DCF schreiben

10037

Ist [DCF schreiben] aktiviert, wird nach dem Einfügen einer EDS-Datei im eingestellten Verzeichnis für Übersetzungsdateien eine DCF-Datei erstellt, deren Namen sich zusammensetzt aus dem Namen der EDS-Datei und dem angehängten Node-ID.

CAN Parameter: Alle SDOs erzeugen

10038

Ist diese Option aktiviert, werden für alle Kommunikationsobjekte SDOs erzeugt.
Default-Werte werden nicht erneut geschrieben!

CAN Parameter: Knoten zurücksetzen

10039

Der Slave wird zurückgesetzt ("load"), sobald die Konfiguration in die Steuerung geladen wird.

CAN Parameter: Optionales Gerät

10040

Ist die Option [Optionales Gerät] aktiviert, versucht der Master nur einmal, von diesem Knoten zu lesen. Bei fehlender Antwort wird der Knoten ignoriert und der Master geht in den normalen Betriebszustand über.

Wird der Slave zu einem späteren Zeitpunkt an das Netzwerk angeschlossen und erkannt, wird er automatisch gestartet. Dazu müssen Sie die Option [Automatisch starten] in den CAN-Parametern des Masters angewählt haben.

CAN Parameter: Nicht initialisieren

10041

Wird diese Option aktiviert, nimmt der Master den Knoten sofort in Betrieb, ohne ihm Konfigurations-SDOs zu schicken. (Die SDO-Daten werden aber dennoch erzeugt und auf der Steuerung gespeichert.)

CAN Parameter: Nodeguarding- / Heartbeat-Einstellungen

10042

Je nach Gerät haben Sie die Wahl:
- [Nodeguarding] und [Life Time Factor] einstellen ODER
- [Heartbeat] einstellen.

Wir empfehlen: Für aktuelle Geräte besser mit Heartbeat arbeiten, weil dann die Buslast niedriger ist.

© ifm electronic gmbh | www.ifm.com

CAN Parameter: Emergency Telegram

10043

Die Option ist im Normalfall angewählt. Die EMCY-Nachrichten werden mit dem angegebenen Identifier übertragen.

CAN Parameter: Communication Cycle

10044

In ganz speziellen Anwendungsfällen können Sie an dieser Stelle eine Überwachungszeit für die vom Master erzeugten SYNC-Nachrichten einstellen.

Bitte beachten Sie, dass diese Zeit länger als die SYNC-Zeit des Masters sein muss. Der optimale Wert muss ggf. experimentell ermittelt werden.

Nodeguarding und Heartbeat reichen in den meisten Fällen zur Knotenüberwachung aus.

© ifm electronic gmbh | www.ifm.com

Register [PDO-Mapping empfangen] und [PDO-Mapping senden]

1969

Die Registerkarten [PDO-Mapping empfangen] und [PDO-Mapping senden] im Konfigurationsdialog eines CAN-Moduls ermöglichen es, dass in der EDS-Datei beschriebene "Mapping" (Zuordnung zwischen lokalem Objektverzeichnis und PDOs vom/zum CANopen-Slave) des Moduls zu verändern (wenn es vom CAN-Modul unterstützt wird).

Auf der linken Seite stehen alle "mapbaren" Objekte der EDS-Datei zur Verfügung und können zu den PDOs (Process Data Objects) der rechten Seite hinzugefügt oder wieder entfernt werden.

Die [StandardDataTypes] können eingefügt werden, um im PDO leere Zwischenräume zu erzeugen.

PDO-Mapping: Einfügen

10046

Mit der Schaltfläche [Einfügen] können Sie weitere PDOs erzeugen und mit entsprechenden Objekten belegen. Über die eingefügten PDOs erfolgt die Zuordnung der Ein- und Ausgänge zu den IEC-Adressen.

In der Steuerungskonfiguration werden die vorgenommenen Einstellungen nach Verlassen des Dialoges sichtbar. Die einzelnen Objekte können dort mit symbolischen Namen belegt werden.

PDO-Mapping: Eigenschaften

10047

Über Eigenschaften lassen sich die in der Norm definierten Eigenschaften der PDOs in einem Dialog editieren:

COB-ID	Jede PDO-Nachricht benötigt einen eindeutigen COB-ID (Communication Object Identifier). Wird eine Option von dem Modul nicht unterstützt oder darf der Wert nicht verändert werden, so erscheint das Feld grau und kann nicht editiert werden.
Inhibit Time	Die Inhibit Time (100 µs) ist die minimale Zeit zwischen zwei Nachrichten dieses PDOs, damit die Nachrichten, die bei Änderung des Wertes übertragen werden, nicht zu häufig versendet werden. Die Einheit ist 100 µs.
Transmission Type	Bei Transmission Type erhalten Sie eine Auswahl von möglichen Übertragungmodi für dieses Modul: **acyclic – synchronous** Das PDO wird nach einer Änderung mit dem nächsten SYNC übertragen. **cyclic – synchronous** Das PDO wird synchron übertragen, wobei [Number of SYNCs] die Anzahl der Synchronisationsnachrichten angibt, die zwischen zwei Übertragungen dieses PDOs liegen. **asynchronous – device specific** Das PDO wird ereignisgesteuert, d.h. wenn sich der Wert ändert, übertragen. Welche Daten auf diese Weise übertragen werden können, ist im Geräteprofil festgelegt. **asynchronous – manufacturer specific** Das PDO wird ereignisgesteuert, d.h. wenn sich der Wert ändert, übertragen. Welche Daten auf diese Weise übertragen werden, wird vom Gerätehersteller festgelegt. **(a)synchronous – RTR only** Diese Dienste sind nicht implementiert. **Number of SYNCs** Abhängig vom Transmission Type ist dieses Feld editierbar zur Eingabe der Anzahl der Synchronisationsnachrichten (Definition in [CAN-Parameter-Dialog], [Com. Cycle Period], [Sync Window Length], [Sync. COB-Id]), nach denen das PDO wieder versendet werden soll. **Event-Time** Abhängig vom Transmission Type wird hier die Zeitspanne in Millisekunden [ms] angegeben, die zwischen zwei Übertragungen des PDOs liegen soll.

© ifm electronic gmbh | www.ifm.com

Register [Service Data Objects]

1970

Index, Name, Wert, Typ und Default

Hier werden alle Objekte der EDS- oder DCF-Datei aufgelistet, die im Bereich von Index 2000_{16} bis $9FFF_{16}$ liegen und als beschreibbar definiert sind. Zu jedem Objekt werden Index, Name, Wert, Typ und Default angegeben. Der Wert kann verändert werden. Markieren Sie den Wert und drücken Sie die [Leertaste]. Nach Änderung können Sie den Wert durch die Taste [Eingabe] bestätigen oder mit [ESC] verwerfen.

Bei der Initialisierung des CAN-Buses werden die eingestellten Werte in Form von SDOs (Service Data Object) an die CAN-Module übertragen und haben damit direkten Einfluss auf das Objektverzeichnis des CANopen-Slaves. Sie werden im Normalfall bei jedem Start des Applikations-Programms neu geschrieben – unabhängig davon, ob sie im CANopen-Slave dauerhaft gespeichert werden.

© ifm electronic gmbh | www.ifm.com

Der Master zur Laufzeit

Inhalt

Reset aller konfigurierten Slaves am Bus beim Systemstart ... 142
Abfrage des Slave-Gerätetyps ... 142
Konfiguration aller fehlerfrei detektierten Geräte ... 143
Automatische Konfiguration von Slaves ... 143
Start aller fehlerfrei konfigurierten Slaves ... 143
Zyklisches Senden der SYNC-Message ... 143
Nodeguarding mit Lifetime-Überwachung ... 143
Heartbeat vom Master an die Slaves ... 143
Empfangen von Emergency-Messages ... 143

8019

Hier lesen Sie über Funktionalität der CANopen-Master-Bibliotheken zur Laufzeit.

Die CANopen-Master-Bibliothek stellt der CoDeSys-Applikation implizite Dienste zur Verfügung, die für die meisten Applikationen ausreichend sind. Diese Dienste werden für den Anwender transparent integriert und stehen in der Applikation ohne zusätzliche Aufrufe zur Verfügung. In der nachfolgenden Beschreibung beschreiben wir die Funktionsbausteine zur Nutzung der Netzwerkdiagnose-, Status- und EMCY-Funktionen. Dafür wird auf die Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB` zugegriffen.

Zu den Diensten der CANopen-Master-Bibliothek zählen:

Reset aller konfigurierten Slaves am Bus beim Systemstart

8020

Um die Slaves zurückzusetzen, wird standardmäßig das NMT-Kommando "Reset Remote Node" benutzt, explizit für jeden Slave einzeln. (NMT steht nach CANopen für **N**etwork **M**anagmen**t**. Die einzelnen Kommandos sind im CAN-Dokument DSP301 beschrieben.) Um Slaves mit weniger leistungsstarken CAN-Controllern nicht zu überlasten, ist es sinnvoll, die Slaves mit einem Kommando "All Remote Nodes" zurückzusetzen.

Alle Slaves zurücksetzen:

► FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) aufrufen:
- NODE = 0
- NMTSERVICE = 3

Einzelnen Slave zurücksetzen:

► FB CANOPEN_NMTSERVICES aufrufen:
- NODE = Knotennummer
- NMTSERVICE = 3

Abfrage des Slave-Gerätetyps

8021

Abfrage des Slave-Gerätetyps mittels SDO (Abfrage des Objekts 1000_{16}) und Vergleich mit dem konfigurierten Slave-ID:

Fehlerstatus-Ausgabe für die Slaves, von denen ein falscher Gerätetyp empfangen wurde. Die Anfrage wird nach 0,5 s wiederholt, wenn:
- kein Gerätetyp wurde empfangen
- UND Slave wurde in der Konfiguration **nicht** als optional markiert
- UND Timeout ist **nicht** abgelaufen.

© ifm electronic gmbh | www.ifm.com

Konfiguration aller fehlerfrei detektierten Geräte

8022

Jedes SDO wird auf Antwort überwacht und wiederholt, wenn sich innerhalb der Überwachungszeit der Slave nicht meldet.

Automatische Konfiguration von Slaves

8023

Automatische Konfiguration von Slaves mittels SDOs bei laufendem Busbetrieb:
Voraussetzung: Der Slave hat sich mittels Bootup-Message beim Master angemeldet.

Start aller fehlerfrei konfigurierten Slaves

8024

Start aller fehlerfrei konfigurierten Slaves nach dem Ende der Konfiguration des betreffenden Slaves:

Zum Starten der Slaves wird normalerweise das NMT-Kommando "Start remote node" benutzt. Wie beim "Reset" kann dieses Kommando durch "Start All Remote Nodes" ersetzt werden.

Alle Knoten starten:

► FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) aufrufen:
- NODE = 0
- NMTSERVICE = 2

Zyklisches Senden der SYNC-Message

8025

Dieser Wert ist nur bei der Konfiguration einstellbar.

Nodeguarding mit Lifetime-Überwachung

8026

Nodeguarding mit Lifetime-Überwachung für jeden Slave einstellbar:

Der Fehlerstatus kann mittels ***CANOPEN_GETGUARDHBERRLIST*** (→ Seite 190) überwacht werden.

Wir empfehlen: Für aktuelle Geräte besser mit Heartbeat arbeiten, weil dann die Buslast niedriger ist.

Heartbeat vom Master an die Slaves

8027

Heartbeat vom Master an die Slaves und überwachen der Heartbeats der Slaves:

Der Fehlerstatus kann mittels ***CANOPEN_GETGUARDHBERRLIST*** (→ Seite 190) überwacht werden.

Empfangen von Emergency-Messages

8028

Empfangen von Emergency-Messages für jeden Slave mit Speicherung der zuletzt empfangenen Emergency-Messages:

Die Fehlernachrichten können mit ***CANOPEN_GETEMCYMESSAGES*** (→ Seite 208) ausgelesen werden.

© ifm electronic gmbh | www.ifm.com

Netzwerk starten

8029

Hier lesen Sie über das Starten des CANopen-Netzwerks.

Nach einem Download des Projekts auf die Steuerung oder einem Reset der Applikation wird das CAN-Netz vom Master neu hochgefahren. Das geschieht immer in der gleichen Reihenfolge von Aktionen:

- Alle Slaves werden zurückgesetzt, außer wenn sie als [nicht initialisieren] im Konfigurator markiert sind. Das Zurücksetzen geschieht einzeln mit dem NMT-Kommando "Reset Node" (81_{16}), jeweils mit dem Node-ID des Slaves. Wurde in ***CANOPEN_SETSTATE*** (→ Seite 178) der Eingang GLOBALSTART gesetzt, wird zum Hochfahren des Netzes das Kommando einmal mit NODEID=0 benutzt.
- Alle Slaves werden konfiguriert. Dazu wird zunächst das Objekt 1000_{16} des Slaves abgefragt.
  - Wenn der Slave innerhalb der Überwachungszeit von 0,5 Sekunden antwortet, wird das jeweils nächste Konfigurations-SDO gesendet.
  - Ist ein Slave als [optional] markiert und antwortet nicht innerhalb der Überwachungszeit auf die Abfrage des Objekts 1000_{16}, wird er als nicht vorhanden markiert und keine weiteren SDOs werden an ihn geschickt.
  - Wenn ein Slave auf die Abfrage des Objekts 1000_{16} mit einem anderen Typ als dem konfigurierten (in den unteren 16 Bit) antwortet, wird er zwar konfiguriert, aber als falscher Typ markiert.
- Alle SDOs werden jeweils solange wiederholt, bis innerhalb einer Überwachungszeit eine Antwort des Slaves gesehen wurde. Mit dem Eingang CFGTIMEOUT des FB CANOPEN_SETSTATE kann eingestellt werden, nach welcher Zeit der Master die Initialisierung abbricht. Mittels ***CANOPEN_GETSTATE*** (→ Seite 181) kann der Status der Initialisierung überprüft werden.
- Wenn der Master eine Heartbeat-Zeit ungleich 0 konfiguriert hat, beginnt die Erzeugung des Hearbeats sofort nach dem Starten der Mastersteuerung.
- Nachdem alle Slaves ihre Konfigurations-SDOs erhalten haben, beginnt für Slaves mit konfiguriertem Nodeguarding das Guarding.
- Wenn der Master auf [automatisch starten] konfiguriert wurde, werden jetzt alle Slaves einzeln vom Master gestartet. Dazu wird das NMT-Kommando "Start Remote Node" (01_{16}) benutzt. Wurde in CANOPEN_SETSTATE der Eingang GLOBALSTART gesetzt, dann wird das Kommando mit Node-ID 0 genutzt und somit alle Slaves mit einem "Start all Nodes" gestartet.
- Es werden mindestens einmal alle konfigurierten TX-PDOs gesendet (für die Slaves sind das RX-PDOs).
- Wenn [automatisch starten] deaktiviert wurde, müssen die Slaves wie folgt gestartet werden:
  - Einzeln starten:
    FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) aufrufen:
    - NODE = Knotennummer
    - NMTSERVICE = 2
  - Global starten:
    FB CANOPEN_NMTSERVICES aufrufen:
    - NODE = 0
    - NMTSERVICE = 2

© ifm electronic GmbH | www.ifm.com

Netzwerkzustände

Inhalt

Hochlauf des CANopen-Masters 145
Hochlauf der CANopen-Slaves 147
Hochlauf des Netzwerks ohne [Automatisch starten] 148
Das Objektverzeichnis des CANopen-Masters 150

8031

Hier lesen Sie, wie Sie die Zustände des CANopen-Netzwerks interpretieren und darauf reagieren können.

Beim ***Netzwerk starten*** (→ Seite 144) des CANopen Netzwerks und während des Betriebs durchlaufen die einzelnen Funktionsblöcke der Bibliothek verschiedene Zustände.

Um den Zugriff zu erleichtern, steht Ihnen ***CANOPEN_GETSTATE*** (→ Seite 181) aus der Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB` zur Verfügung.

Hochlauf des CANopen-Masters

8469

Während des Hochlaufs des CAN-Netzwerks durchläuft der Master verschiedene Zustände, die Sie über den Ausgang MASTERSTATE in ***CANOPEN_GETSTATE*** (→ Seite 181) ablesen können. (Netzwerk-Status des Masters → nächstes Kapitel)

> Immer, wenn ein Slave auf eine SDO-Anfrage (Upload oder Download) nicht antwortet, dann wird die Anfrage wiederholt.

> Der Master verlässt den Status 3, wie oben beschrieben, aber erst, wenn alle SDOs erfolgreich übertragen wurden oder der in ***CANOPEN_NMTSERVICES*** (→ Seite 175) eingestellte TIMEOUT abgelaufen ist.

> Über den Ausgang NMTSTATUS des FB CANOPEN_NMTSERVICES kann erkannt werden, ob ein Slave fehlt oder ob der Master nicht alle SDOs richtig empfangen kann.

> Der Master wechselt in den Status 5 unabhängig davon, ob ein Slave mit einer Bestätigung oder einem Abort antwortet. Für den Master ist nur von Interesse, ob er überhaupt eine Antwort empfangen hat.

Eine Ausnahme stellt ein als [optional] markierter Slave dar.

> Optionale Slaves werden nur einmal nach ihrem Objekt 1000_{16} gefragt. Wenn sie nicht innerhalb von 0,5 s antworten, wird der Slave vom Master zunächst ignoriert und der Master geht auch ohne weitere Reaktion dieses Slaves in Status 5.

© ifm electronic gmbh | www.ifm.com

NMT-Status für CANopen-Master

9964

Status hex	dez	Beschreibung
00	0	nicht definiert
01	1	Master wartet auf die Bootup-Nachricht des Slaves. ODER: Master wartet auf Ablauf der GuardTime.
02	2	- Master wartet 300 ms. - Master fordert das Objekt 1000_{16} an. - Danach wechselt der Master auf Status 3.
03	3	Der Master konfiguriert seine Slaves. Dazu sendet der Master an die Slaves der Reihe nach alle vom Konfigurator erzeugten SDOs: - Der Master sendet an den Slave ein SDO-Read-Request (Index 1000_{16}). - Die generierten SDOs werden in ein SDO-Array gepackt. - Der Slave kennt seine erste SDO und die Anzahl seiner SDOs.
05	5	Nachdem an alle Slaves die SDOs übertragen wurden, geht der Master in den Status 5 und bleibt in diesem Status. Status 5 ist für den Master der normale Betriebszustand.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Ausgang NODE_STATE
CANOPEN_GETSTATE	Ausgang NODESTATE

© ifm electronic gmbh | www.ifm.com

Hochlauf der CANopen-Slaves

8032

Die Status eines Slaves können Sie über die Ausgänge des FB ***CANOPEN_GETSTATE*** (→ Seite 181) auslesen.
(Netzwerk-Status der Slaves → nächstes Kapitel)

NMT-Status für CANopen-Slave

9965

Status hex	dez	Beschreibung
FF	-1	Der Slave wird durch die NMT-Nachricht [Reset Node] zurückgesetzt und wechselt selbständig in den Status 1.
00	0	nicht definiert
01	1	Status = Warten auf BOOTUP Der Slave wechselt nach einer maximalen Zeit von 2 s oder sofort nach Empfang seiner Bootup-Message in den Status 2.
02	2	Status = BOOTUP Der Slave wechselt nach einer Verzögerungszeit von 0,5 s automatisch in den Status 3.
03	3	Status = PREPARED Im Status 3 wird der Slave konfiguriert. Der Slave bleibt solange im Status 3, bis er alle vom Konfigurator erzeugten SDOs erhalten hat. Dabei spielt es keine Rolle, ob während der Konfiguration vom Slave SDO-Transfers mit Abort (Fehler) oder ob alle fehlerfrei beantwortet wurden. Nur die vom Slave erhaltene Antwort als solche ist wichtig – nicht ihr Inhalt. Wenn im Konfigurator die Option [Knoten zurücksetzen] aktiviert wurde, wird nach dem Senden des Objekts 1011_{16} Subindex 1, der dann den Wert "load" enthält, ein erneuter Reset des Slaves durchgeführt. Der Slave wird dann wieder mit dem Upload des Objekts 1000_{16} angefragt. Slaves, bei denen während der Konfigurationsphase ein Problem auftritt, bleiben im Status 3 oder wechseln nach der Konfigurationsphase direkt in einen Fehlerstatus (Status > 5).
04	4	Status = PRE-OPERATIONAL Ein Knoten wechselt immer in den Status 4, außer: • es handelt sich um einen "optionalen" Slave und er wurde als nicht am Bus verfügbar detektiert (Abfrage Objekt 1000_{16}) ODER: • der Slave ist zwar vorhanden, aber hat auf die Abfrage des Objekts 1000_{16} mit einem anderen Typ in den unteren 16 Bits reagiert, als der Konfigurator erwartet hat.
05	5	Status = OPERATIONAL Im Status 5 findet der normale Datenaustausch statt: "Normal Operation". Wenn der Master auf [Automatisch starten] konfiguriert wurde, wird der Slave im Status 4 gestartet (d.h. es wird eine "Start Node"-NMT-Nachricht erzeugt) und der Slave wechselt automatisch nach Status 5. Wurde GLOBAL_START gesetzt, dann wird gewartet, bis sich alle Slaves im Status 4 befinden. Anschließend werden alle Slaves mit dem NMT-Kommando [Start All Nodes] gestartet.
61	97	Ein Knoten wechselt in den Status 97, wenn er optional ist (optionales Gerät in der CAN-Konfiguration) und nicht auf die SDO-Anfrage nach dem Objekt 1000_{16} reagiert hat. Wird der Slave zu einem späteren Zeitpunkt an das Netzwerk angeschlossen und erkannt, wird er automatisch gestartet. Dazu müssen Sie aber die Option [Automatisch starten] in den CAN-Parametern des Masters angewählt haben.
62	98	Ein Knoten wechselt in den Status 98, wenn der Gerätetyp (Objekt 1000_{16}) nicht dem konfigurierten Typ entspricht.
63	99	Im Falle eines Nodeguarding-Timeouts wird der Slave auf Status 99 gesetzt. Sobald der Slave wieder auf NodeGuard-Anfragen reagiert und die Option [Automatisch starten] eingeschaltet ist, wird er automatisch vom Master gestartet. Dabei wird der Knoten abhängig von seinem Status, der in der Antwort auf die Nodeguard-Anfragen enthalten ist, neu konfiguriert oder nur gestartet. Um den Slave manuell zu starten, genügt es, die Methode [NodeStart] zu benutzen.

© ifm electronic gmbh | www.ifm.com

Der Master sendet Nodeguard-Nachrichten an den Slave, ...
- wenn sich der Slave im Status 4 oder höher befindet UND
- wenn Nodeguarding konfiguriert wurde.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Ausgang NODE_STATE
CANOPEN_GETSTATE	Ausgang NODESTATE

CANopen-Status des Knotens

1973

Knotenstatus nach CANopen (mit diesen Werten wird der Status auch in den entsprechenden Nachrichten vom Knoten her codiert).

Status hex	dez	CANopen-Status	Beschreibung
00	0	BOOTUP	Knoten hat die BOOTUP-Nachricht erhalten.
04	4	PREPARED	Knoten wird per SDOs konfiguriert.
05	5	OPERATIONAL	Knoten nimmt am normalen Datenaustausch teil.
7F	127	PRE-OPERATIONAL	Knoten sendet keine Daten, ist aber vom Master konfigurierbar.

Wenn Nodeguarding aktiv: das höchstwertige Status-Bit wechselt (toggelt) von Nachricht zu Nachricht.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Strukturelement LAST_STATE aus dem Array NODE_STATE_SLAVE
CANOPEN_GETSTATE	Ausgang LASTNODESTATE

© ifm electronic gmbh | www.ifm.com

Hochlauf des Netzwerks ohne [Automatisch starten]

Inhalt

Starten des Netzwerks mit GLOBALSTART 149
Starten aller Knoten des Netzwerks mit CAN-Telegramm 149
Rücksetzen aller Knoten des Netzwerks mit CAN-Telegramm 149
Zugriff auf den Status des CANopen-Masters 150

8034

Manchmal ist es notwendig, dass die Applikation den Zeitpunkt bestimmt, wann die CANopen-Slaves gestartet werden. Dazu müssen Sie den FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) verwenden.

Starten des Netzwerks mit GLOBALSTART

8035

In einem CAN-Netz mit vielen Teilnehmern (meist mehr als 8) kommt es häufig dazu, dass schnell aufeinanderfolgende NMT-Nachrichten nicht von allen (meist langsamen) IO-Knoten (z.B. CompactModule CR2013) erkannt werden. Das liegt daran, dass diese Knoten alle Nachrichten mit dem ID 0 mithören müssen. In zu schneller Folge gesendete NMT-Nachrichten überlasten den Empfangspuffer solcher Knoten.

Eine Abhilfe können Sie schaffen, wenn die Anzahl schnell aufeinanderfolgender NMT-Nachrichten reduziert wird.

► FB ***CANOPEN_SETSTATE*** (→ Seite 178) aufrufen:
- GLOBALSTART = TRUE (mit [Automatisch starten])

> Die CANopen-Master-Bibliothek benutzt den Befehl "Start All Nodes", anstatt alle Knoten einzeln mit dem Kommando "Start Node" zu starten.

> GLOBALSTART wird nur einmalig bei der Netzwerk-Initialisierung ausgeführt.

> Wenn dieser Eingang gesetzt wird, startet die Steuerung auch Knoten mit dem Status 98 (siehe oben). Die PDOs für diese Nodes bleiben jedoch weiterhin deaktiviert.

Starten aller Knoten des Netzwerks mit CAN-Telegramm

8036

Wird das Netzwerk nicht automatisch mit GLOBALSTART des FB ***CANOPEN_SETSTATE*** (→ Seite 178) gestartet, kann es jederzeit gestartet werden, d.h. jeder Knoten einzeln nacheinander. Ist das nicht gewünscht, besteht folgende Möglichkeit:

► FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) aufrufen:
- NODE = 0
- NMTSERVICE = 2

> Der Aufruf wird typisch zur Laufzeit durch das Applikations-Programm ausgeführt.

> Bei diesem Aufruf werden auch Knoten mit dem Status 98 (siehe oben) gestartet. Die PDOs für diese Knoten bleiben jedoch weiterhin deaktiviert.

Rücksetzen aller Knoten des Netzwerks mit CAN-Telegramm

8037

Aus denselben Gründen, die für ein NMT-Kommando zum Starten aller Knoten in einem Netz sprechen, gibt es Fälle, in denen Sie besser ein NMT-Kommando zum Rücksetzen aller Knoten im Netz (anstelle eines NMT-Kommandos für jeden einzelnen Knoten) einsetzen.

► FB ***CANOPEN_NMTSERVICES*** (→ Seite 175) aufrufen:
- NODE = 0
- NMTSERVICE = 3

> Dadurch werden einmalig alle Knoten gleichzeitig zurückgesetzt.

© ifm electronic gmbh | www.ifm.com

Zugriff auf den Status des CANopen-Masters

8038

Damit der Applikations-Code erst abgearbeitet wird, wenn das IO-Netzwerk bereit ist, sollten Sie den Status des Masters abfragen. Das folgende Code-Fragment-Beispiel zeigt eine Möglichkeit:

Variablendeklaration

```
VAR
    FB_MasterStatus : CANOPEN_GETSTATE;
    ...
END_VAR
```

Programmcode

```
IF   FB_MasterStatus.MASTERSTATE = 5 THEN
     <Applikationscode>
END_IF
```

Durch Setzen eines Wertes für den Eingang NODE der Funktion ***CANOPEN_GETSTATE*** (→ Seite 181) kann die Applikation reagieren und zum Beispiel den nicht konfigurierbaren Knoten überspringen.

Das Objektverzeichnis des CANopen-Masters

8039

In manchen Fällen ist es hilfreich, wenn der CANopen-Master über ein eigenes Objektverzeichnis verfügt. Das ermöglicht z.B. den Datenaustausch der Applikation mit anderen CAN-Knoten.

Das Objektverzeichnis des Masters wird über eine EDS-Datei mit dem Namen `CRnnnnMasterODEntry.EDS` während der Übersetzungszeit erstellt und mit Werten vorbelegt. Diese EDS-Datei ist im Verzeichnis `CoDeSys Vn\Library\PLCconf` abgelegt. Der Inhalt der EDS-Datei kann über die Schaltfläche [EDS...] im Konfigurations-Fenster [CAN-Parameter] angesehen werden.

Auch, wenn das Objektverzeichnis nicht vorhanden ist, kann der Master ohne Einschränkungen genutzt werden.

Der Zugriff auf das Objektverzeichnis durch die Applikation erfolgt über Funktionsblöcke:
- ***CANOPEN_GETODCHANGEDFLAG*** (→ Seite 194),
- ***CANOPEN_READOBJECTDICT*** (→ Seite 195),
- ***CANOPEN_WRITEOBJECTDICT*** (→ Seite 196).

An der Oberfläche verfügt CoDeSys über keinen Editor für dieses Objektverzeichnis.

Die EDS-Datei gibt nur vor, mit welchen Objekten das Objektverzeichnis angelegt wird.

Wenn ein Objektverzeichnis im Master vorhanden ist, kann der Master als SDO-Server im Netz auftreten. Immer wenn ein Client auf einen Objektverzeichnis-Eintrag schreibend zugreift, wird das der Applikation über den Ausgang ODCHANGED des FB CANOPEN_GETODCHANGEDFLAG angezeigt. Nach der Auswertung müssen Sie dieses Flag durch Setzen des Eingangs RESETFLAG wieder zurücksetzen.

Die Applikation kann das Objektverzeichnis nutzen, indem die Einträge direkt beschrieben oder gelesen werden.

© ifm electronic gmbh | www.ifm.com

CANopen-Slave

Inhalt

Funktionalität der CANopen-Slave-Bibliothek 151
CANopen-Slave konfigurieren 151
Zugriff auf den CANopen-Slave zur Laufzeit 157

1865

Eine CoDeSys-programmierbare Steuerung kann in einem CAN-Netzwerk auch als CANopen-Slave erscheinen.

Funktionalität der CANopen-Slave-Bibliothek

8041

Die CANopen-Slave-Bibliothek zusammen mit dem CANopen-Konfigurator stellt dem Anwender folgende Möglichkeiten zur Verfügung:

- In CoDeSys Konfiguration der Eigenschaften NodeGuarding/Heartbeat, Emergency, Node-ID und Baudrate, auf der das Gerät arbeiten soll.
- Zusammen mit dem Parametermanager in CoDeSys kann ein Default-PDO-Mapping erstellt werden, das zur Laufzeit vom Master geändert werden kann. Die Änderung des PDO-Mappings erfolgt während der Konfigurationsphase durch den Master. Durch das Mapping können IEC-Variablen der Applikation in PDOs gemappt werden. D.h. den PDOs werden IEC-Variable zugeordnet, um sie im Applikations-Programm einfach auswerten zu können.
- Die CANopen-Slave-Bibliothek stellt ein Objektverzeichnis zur Verfügung. Die Größe dieses Objektverzeichnisses wird zur Übersetzungszeit von CoDeSys festgelegt. In diesem Verzeichnis befinden sich alle Objekte, die den CANopen-Slave beschreiben und zusätzlich die, die vom Parametermanager definiert sind. Im Parametermanager können zusammen mit dem CANopen-Slave nur die Listenarten Parameter und Variablen verwendet werden.
- Die Bibliothek verwaltet die Zugriffe auf das Objektverzeichnis, tritt also am Bus als SDO-Server auf.
- Die Bibliothek überwacht das Nodeguarding und die Heartbeat-Consumer-Zeit (immer nur von einem Producer) und setzt entsprechende Fehlerflags für die Applikation.
- Es kann eine EDS-Datei erzeugt werden, die die konfigurierten Eigenschaften des CANopen-Slave so beschreibt, dass das Gerät als Slave unter einem CANopen-Master eingebunden und konfiguriert werden kann.

Die CANopen-Slave Bibliothek stellt ausdrücklich folgende, in CANopen beschriebene, Funktionalitäten nicht zur Verfügung (alle hier und im obigen Abschnitt nicht genannten Möglichkeiten des CANopen-Protokolls sind ebenfalls nicht implementiert):

- Dynamische SDO- und PDO-Identifier
- Automatische Erzeugung von Emergency-Nachrichten, mit Ausnahme der Guarding/Heartbeat Fehler:
  Emergency-Nachrichten in der Applikation erzeugen mit ***CANOPEN_SENDEMCYMESSAGE*** (→ Seite 205).
  Emergency-Nachrichten in der Applikation lesen mit ***CANOPEN_GETEMCYMESSAGES*** (→ Seite 208).
  Den Baustein / die Bausteine finden Sie in der Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`.
- Dynamische Änderungen der PDO-Eigenschaften werden z.Z. immer nur beim Eintreffen einer StartNode NMT-Nachricht übernommen, nicht mit den in CANopen definierten Mechanismen.

© ifm electronic gmbh | www.ifm.com

CANopen-Slave konfigurieren

Inhalt

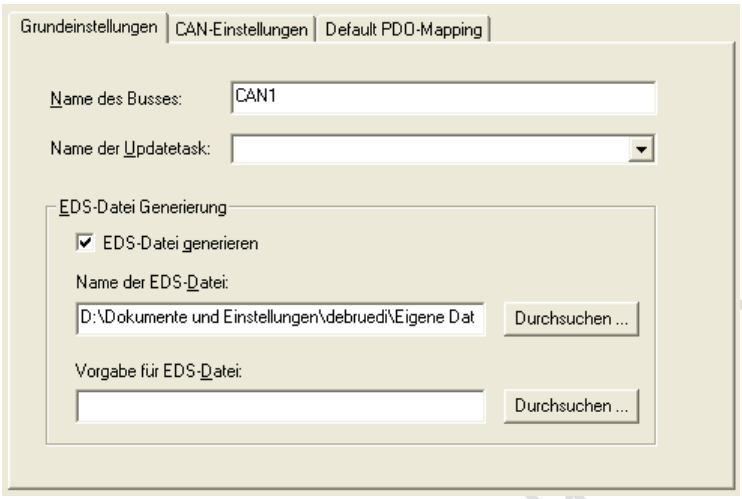
Register [Grundeinstellungen] 152
Register [Default PDO-Mapping] 154
Verändern des Standard-Mappings durch Master-Konfiguration 156

7707

Um die Steuerung als CANopen-Slave zu nutzen, muss zunächst in der Steuerungskonfiguration über [Einfügen] > [Unterelement anhängen] der CANopen-Slave an der gewünschten CAN-Schnittstelle eingefügt werden. Alle notwendigen Bibliotheken werden automatisch in den Bibliotheksverwalter eingefügt.

Register [Grundeinstellungen]

1981



Grundeinstellungen: Name des Busses

10049

Parameter wird im Moment nicht benutzt.

Grundeinstellungen: Name der Updatetask

10050

Name der Task, in der der Aufruf des CANopen-Slave erfolgt.

Grundeinstellungen: EDS-Datei generieren

10051

Soll aus den Einstellungen hier eine EDS-Datei erzeugt werden, um den CANopen-Slave in eine beliebigen Masterkonfiguration einfügen zu können, muss hier die Option [EDS-Datei generieren] aktiviert werden und der Name einer Datei angegeben werden. Optional kann auch noch eine Vorlagendatei angeben werden, deren Einträge zum EDS-File des CANopen-Slave hinzugefügt werden. Bei Überschneidungen werden Vorgaben der Vorlage nicht überschrieben.

© ifm electronic gmbh | www.ifm.com

Beispiel für ein Objektverzeichnis

1991

Folgende Einträge könnten zum Beispiel im Objektverzeichnis stehen:

```
[FileInfo]
FileName=D:\CoDeSys\lib2\plcconf\MyTest.eds
FileVersion=1
FileRevision=1
Description=EDS for CoDeSys-Project:
D:\CoDeSys\CANopenTestprojekte\TestHeartbeatODsettings_Device.pro
CreationTime=13:59
CreationDate=09-07-2005
CreatedBy=CoDeSys
ModificationTime=13:59
ModificationDate=09-07-2005
ModifiedBy=CoDeSys

[DeviceInfo]
VendorName=3S Smart Software Solutions GmbH
ProductName=TestHeartbeatODsettings_Device
ProductNumber=0x33535F44
ProductVersion=1
ProductRevision=1
OrderCode=xxxx.yyyy.zzzz
LMT_ManufacturerName=3S GmbH
LMT_ProductName=3S_Dev
BaudRate_10=1
BaudRate_20=1
BaudRate_50=1
BaudRate_100=1
BaudRate_125=1
BaudRate_250=1
BaudRate_500=1
BaudRate_800=1
BaudRate_1000=1
SimpleBootUpMaster=1
SimpleBootUpSlave=0
ExtendedBootUpMaster=1
ExtendedBootUpSlave=0

...

[1018sub0]
ParameterName=Number of entries
ObjectType=0x7
DataType=0x5
AccessType=ro
DefaultValue=2
PDOMapping=0

[1018sub1]
ParameterName=VendorID
ObjectType=0x7
DataType=0x7
AccessType=ro
DefaultValue=0x0
PDOMapping=0

[1018sub2]
ParameterName=Product Code
ObjectType=0x7
DataType=0x7
AccessType=ro
DefaultValue=0x0
PDOMapping=0
```

Bedeutung der einzelnen Objekte entnehmen Sie bitte der CANopen-Spezifikation DS301.

© ifm electronic gmbh | www.ifm.com

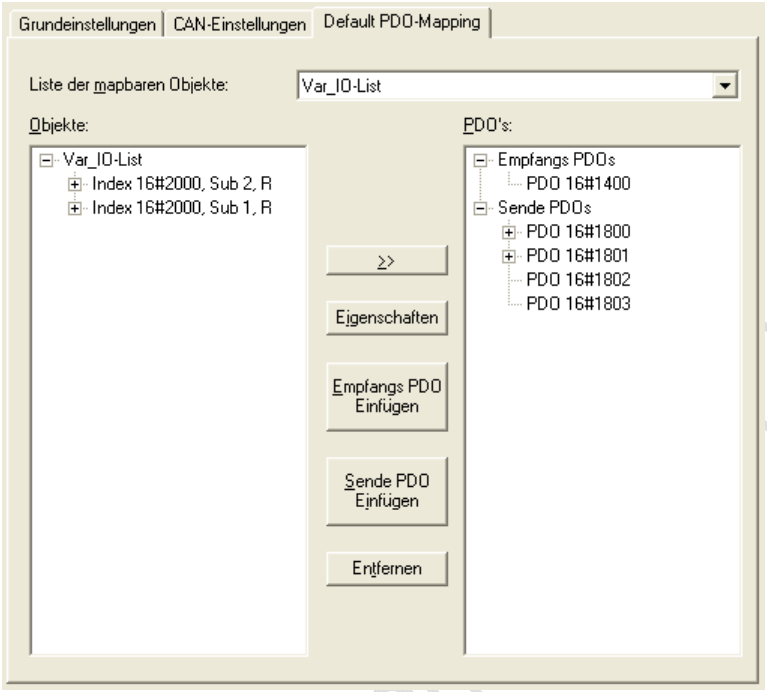
Die EDS-Datei enthält, neben den vorgeschriebenen Einträgen, die Definitionen für SYNC, Guarding, Emergency und Heartbeat. Wenn diese Objekte nicht benutzt werden, sind die Werte auf 0 gesetzt (voreingestellt). Da die Objekte aber im Objektverzeichnis des Slaves zur Laufzeit vorhanden sind, werden sie in der EDS-Datei auch beschrieben.

Das Gleiche gilt für die Einträge für die Kommunikations- und Mapping-Parameter. Es sind immer alle 8 möglichen Subindizes der Mapping-Objekte $16xx_{16}$ oder $1Axx_{16}$ vorhanden, aber u.U. im Subindex 0 nicht berücksichtigt.

Bit-Mapping wird von der Bibliothek nicht unterstützt!

Register [Default PDO-Mapping]

1983



In diesem Register können Sie die Zuordnung zwischen lokalem Objektverzeichnis (OD-Editor) und den PDOs festlegen, die vom CANopen-Slave gesendet/empfangen werden. Eine solche Zuordnung wird als "Mapping" bezeichnet.

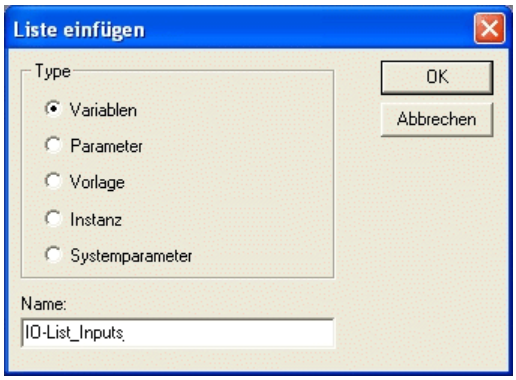
In den verwendeten Objektverzeichniseinträgen (Variablen OD) wird zwischen Objektindex/Subindex die Verbindung zu Variablen der Applikation hergestellt. Dabei müssen Sie nur darauf achten, dass der Subindex 0 eines Indexes, der mehr als einen Subindex enthält, die Information über die Anzahl der Subindizes enthält.

© ifm electronic / www.ifm.com

Beispiel: Variablenliste

10052

Auf dem ersten Empfangs-PDO (COB-ID = 512 + Node-ID) des CANopen-Slaves sollen die Daten für die Variable PLC_PRG.a empfangen werden.



Info

Als Listentyp kann [Variablen] oder [Parameter] gewählt werden.

Zum Datenaustausch (z.B. über PDOs oder sonstige Einträge im Objektverzeichnis) wird eine Variablenliste angelegt.

Die Parameterliste sollten Sie einsetzen, wenn Sie Objektverzeichniseinträge nicht mit Applikations-Variablen verknüpfen wollen. Für die Parameterliste ist zurzeit nur der Index 1006_{16} / SubIdx 0 vordefiniert. In diesen Eintrag kann vom Master der Wert für die [Com. Cycle Period] eingetragen werden. Damit wird das Ausbleiben der SYNC-Nachricht gemeldet.

Also müssen Sie im Objektverzeichnis (Parametermanager) eine Variablenliste anlegen und einen Index/SubIndex mit der Variablen PLC_PRG.a verknüpfen:

► Dazu fügen Sie in der Variablenliste eine Zeile hinzu (rechte Maustaste öffnet das Kontextmenü) und tragen einen Variablen-Namen (beliebig) sowie den Index und den Subindex ein.

► Als Zugriffsrichtung ist für ein Empfangs-PDO nur [write only] (schreiben) zugelassen.

► In die Spalte [Variable] tragen Sie dann "`PLC_PRG.a`" ein, oder drücken [F2] und wählen die Variable aus.

HINWEIS

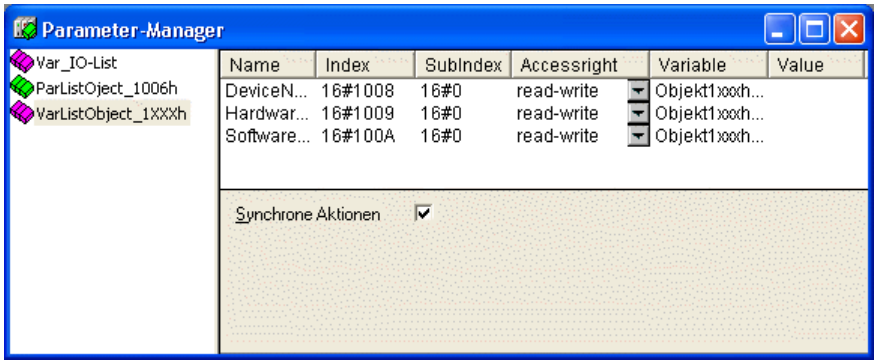
Daten, die vom CANopen-Master gelesen werden sollen (z.B. Eingänge, Systemvariablem), müssen die Zugriffsrichtung (Accessright) [read only] (lesen) haben.

Daten, die vom CANopen-Master geschrieben werden sollen (z.B. Ausgänge im Slave), müssen die Zugriffsrichtung (Accessright) [write only] (schreiben) haben.

SDO-Parameter, die vom CANopen-Master geschrieben und gleichzeitig aus der Slave-Applikation gelesen und geschrieben werden sollen, müssen die Zugriffsrichtung (Accessright) [read-write] (lesen+schreiben) haben.

© ifm electronic gmbH | www.ifm.com

Damit Sie den Parametermanager öffnen können, muss in den Zielsystemeinstellungen unter [Netzfunktionen] der Parametermanager aktiviert sein. Die Bereiche für Index/Subindex sind bereits mit sinnvollen Werten vorbelegt und sollten nicht geändert werden.



Im Default PDO-Mapping des CANopen-Slaves wird anschließend der Index-/Subindex-Eintrag als Mapping-Eintrag einem Empfangs-PDO zugewiesen. Die Eigenschaften des PDOs lassen sich über den Dialog festlegen, der aus Kapitel ***CANopen-Slaves einfügen und konfigurieren*** (→ Seite 137) bekannt ist.

Nur Objekte aus dem Parametermanager, die mit dem Attribut [read only] (lesen) oder [write only] (schreiben) versehen sind, werden in der evtl. erzeugten EDS-Datei als mapbar (= zuordnungsfähig) markiert und tauchen in der Liste der mapbaren Objekte auf. Alle anderen Objekte werden in der EDS-Datei als nicht mapbar markiert.

HINWEIS

Werden mehr als 8 Datenbytes in ein PDO gemappt, werden automatisch die nächsten freien Identifier dafür genutzt, bis alle Datenbytes übertragen werden können.

Um eine klare Struktur der verwendeten Identifier zu erhalten, sollten Sie die richtige Zahl der Empfangs- und Sende-PDOs einfügen und diesen die Variablen-Bytes aus der Liste zuordnen.

Verändern des Standard-Mappings durch Master-Konfiguration

1984

Sie können das vorgegebene PDO-Mapping (in der CANopen-Slave-Konfiguration) in bestimmten Grenzen durch den Master verändern.

Dabei gilt die Regel, dass der CANopen-Slave nicht in der Lage ist, Objektverzeichniseinträge neu anzulegen, die nicht bereits im Standard-Mapping (Default PDO-Mapping in der CANopen-Slave-Konfiguration) vorhanden sind. Also kann z.B. für ein PDO, das im Default PDO-Mapping ein gemapptes Objekt enthält, in der Masterkonfiguration kein zweites Objekt gemappt werden.

Das durch die Masterkonfiguration veränderte Mapping kann also höchstens die im Standard-Mapping vorhandenen PDOs enthalten. Innerhalb dieser PDOs sind 8 Mapping-Einträge (Subindizes) vorhanden.

Eventuelle Fehler, die hierbei auftreten können, werden Ihnen nicht angezeigt, d.h. die überzähligen PDO-Definitionen / die überzähligen Mapping-Einträge werden so behandelt, als seien sie nicht vorhanden.

Die PDOs müssen im Master immer von 1400_{16} (Empfangs-PDO-Kommunikationsparameter) oder 1800_{16} (Sende-PDO-Kommunikationsparameter) beginnend angelegt sein und lückenlos aufeinander folgen.

© ifm electronic gmbh | www.ifm.com

Zugriff auf den CANopen-Slave zur Laufzeit

1985

Einstellen der Knotennummer und der Baud-Rate eines CANopen-Slave

8042

Beim CANopen-Slave kann zur Laufzeit des Applikations-Programms die Knotennummer und die Baudrate eingestellt werden.

Einstellen der Knotennummer	***CANOPEN_SETSTATE*** (→ Seite 178)
Einstellen der Baud-Rate	***CANOPEN_ENABLE*** (→ Seite 164)

Den Baustein / die Bausteine finden Sie in der Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`.

Zugriff auf die OD-Einträge vom Applikations-Programm

8043

Standardmäßig gibt es Objektverzeichniseinträge, die auf Variablen gemappt sind (Parametermanager).

Es gibt jedoch auch die automatisch erzeugten Einträge des CANopen-Slave, auf die Sie nicht über den Parametermanager in einen Variableninhalt mappen können. Diese Einträge stehen hier zur Verfügung:
- ***CANOPEN_READOBJECTDICT*** (→ Seite 195)
- ***CANOPEN_WRITEOBJECTDICT*** (→ Seite 196)

Den Baustein / die Bausteine finden Sie in der Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`.

Ändern der PDO-Eigenschaften zur Laufzeit

1988

Sollen die Eigenschaften eines PDOs zur Laufzeit verändert werden, so funktioniert das durch einen anderen Knoten über SDO-Schreibzugriffe, wie dies von CANopen beschrieben wird.

Alternativ kann man auch direkt eine neue Eigenschaft, wie z.B. die "Event time" eines Sende-PDOs schreiben und anschließend einen Befehl "StartNode-NMT" an den Knoten schicken, obwohl er bereits gestartet ist. Das führt dazu, dass das Device die Werte im Objektverzeichnis neu interpretiert.

Emergency-Messages durch das Applikations-Programm senden

8044

Zum Senden einer Emergency-Message durch das Applikations-Programm, ***CANOPEN_SENDEMCYMESSAGE*** (→ Seite 205) einsetzen.

Den Baustein / die Bausteine finden Sie in der Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`.

© ifm electronic gmbh | www.ifm.com

CANopen-Netzwerkvariablen

Inhalt

Allgemeine Informationen 158
CANopen-Netzwerkvariablen konfigurieren 158
Besonderheiten bei Netzwerkvariablen 163

1868

Allgemeine Informationen

2076

Netzwerkvariablen

CAN Netzwerkvariablen sind eine Möglichkeit, Daten zwischen zwei oder mehreren Steuerungen auszutauschen. Der Mechanismus sollte dabei für den Anwender möglichst einfach zu handhaben sein. Derzeit sind Netzwerkvariablen auf Basis von CAN und UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. In UDP sind diese als Broadcast-Telegramme realisiert, in CAN als PDOs. Diese Dienste sind vom Protokoll her nicht bestätigte Dienste, d.h. es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

Objektverzeichnis

Das Objektverzeichnis ist eine weitere Möglichkeit, Variablen auszutauschen. Dabei handelt es sich um eine 1-zu-1-Verbindung, die ein bestätigtes Protokoll verwendet. Hier kann der Anwender also kontrollieren, ob die Nachricht den Empfänger erreichte. Der Austausch erfolgt nicht automatisch, sondern über den Aufruf von Funktionsblöcken aus dem Applikations-Programm.
→ Kapitel ***Das Objektverzeichnis des CANopen-Masters*** (→ Seite 150)

CANopen-Netzwerkvariablen konfigurieren

Inhalt

Einstellungen in den Zielsystemeinstellungen 159
Einstellungen in den globalen Variablenlisten 160

8518

Um die Netzwerkvariablen mit CoDeSys zu nutzen, benötigen Sie die Bibliotheken
- `3S_CanDrv.lib`,
- `3S_CANopenManager.lib`,
- `3S_CANopenNetVar.lib` und
- `ifm_NetVarLib_NT_Vxxyyzz.LIB`.

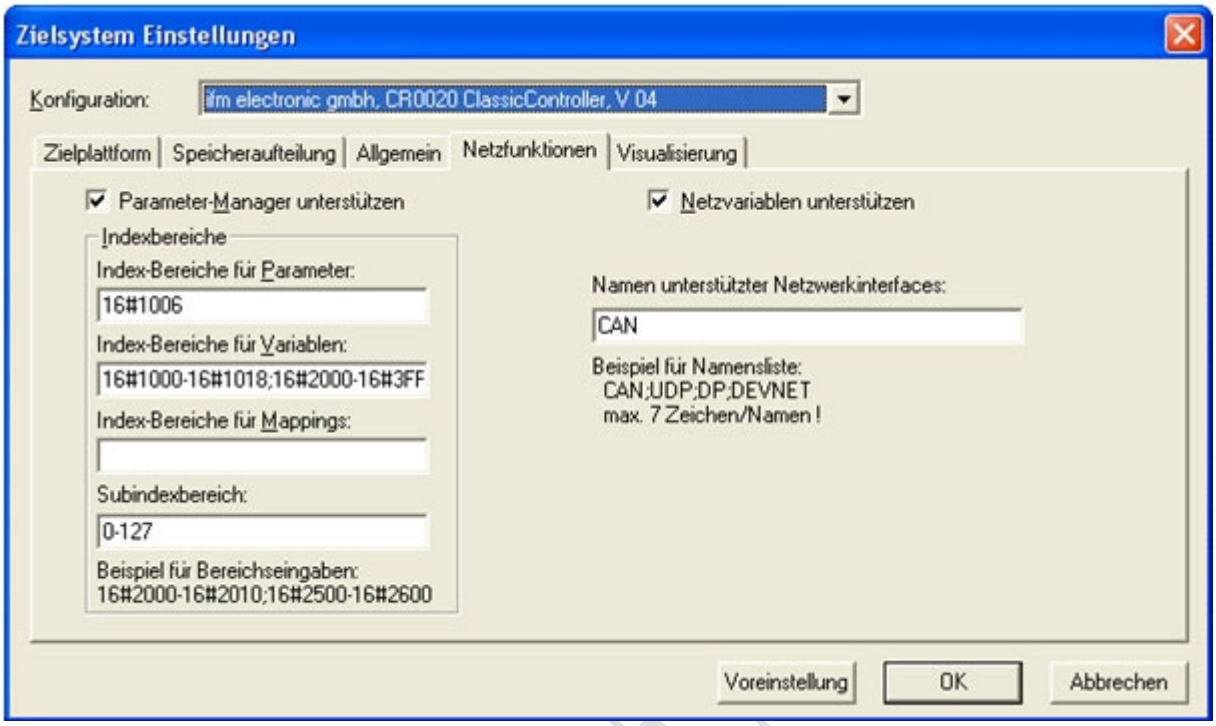
aus dem Verzeichnis `\Targets\ifm\Library\ifm_CANopen`.
Außerdem benötigen Sie die Bibliothek `SysLibCallback.lib`.

CoDeSys erzeugt automatisch den nötigen Initialisierungscode sowie den Aufruf der Netzwerk-Bausteine am Zyklusanfang und -ende.

© ifm electronic gmbh · www.ifm.com

Einstellungen in den Zielsystemeinstellungen

1994



Beispiel: Zielsystemeinstellungen für ClassicController CR0020

► Dialogbox [Zielsystemeinstellungen] wählen

► Register [Netzfunktionen] wählen

► Aktivieren Sie das Kontrollkästchen [Netzvariablen unterstützen].

► Bei [Namen unterstützter Netzwerkinterfaces] geben Sie den Namen des gewünschten Netzwerks an, hier CAN.

► Um Netzwerkvariablen zu nutzen, müssen Sie außerdem einen CANopen-Master oder CANopen-Slave in der Steuerungskonfiguration einfügen.

► Bitte beachten Sie die Besonderheiten bei der Anwendung von Netzwerkvariablen für die jeweiligen Gerätetypen
→ Kapitel *Besonderheiten bei Netzwerkvariablen* (→ Seite 163)

© ifm electronic gmbh

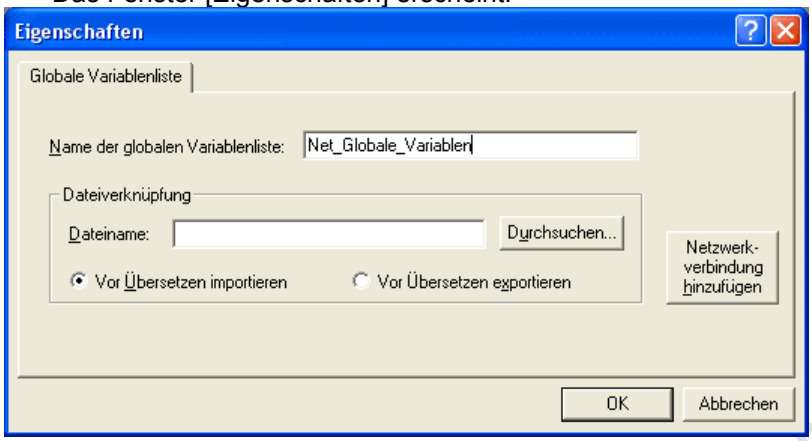
Einstellungen in den globalen Variablenlisten

1995

► Legen Sie eine neue globale Variablenliste an. Hier definieren Sie die Variablen, die sie mit anderen Steuerungen austauschen wollen.

► Öffnen Sie den Dialog mit dem Kontextmenü [Objekt Eigenschaften...].

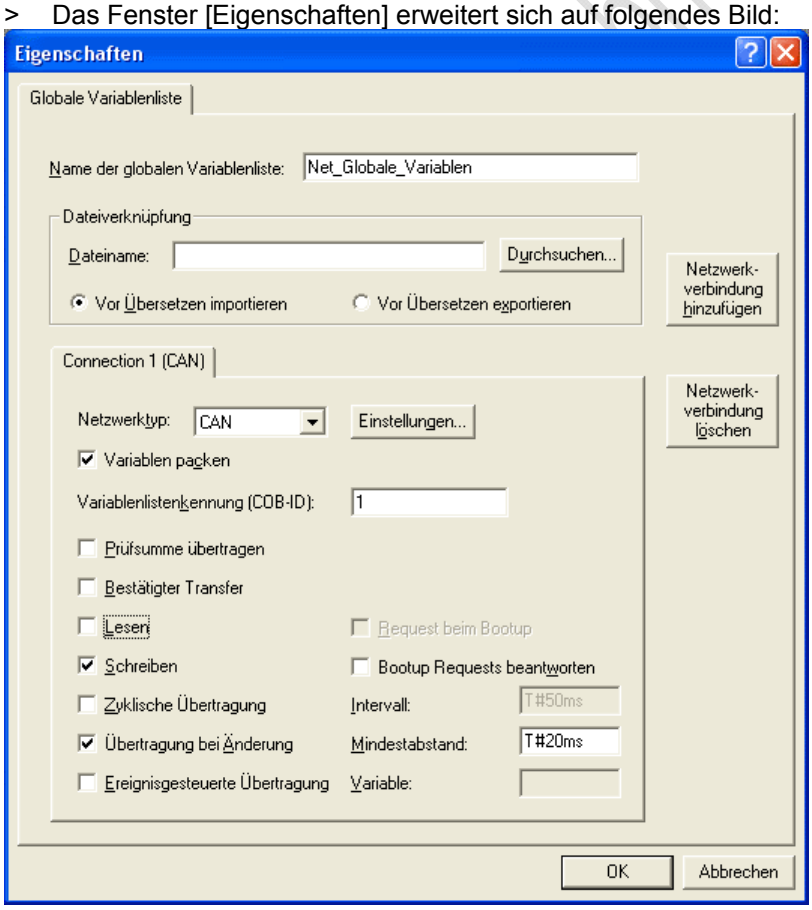
> Das Fenster [Eigenschaften] erscheint:



Wenn Sie die Netzwerkeigenschaften dieser Variablenliste definieren wollen:

► Schaltfläche [Netzwerkverbindung hinzufügen] klicken.
Wenn Sie mehrere Netzwerkverbindungen konfiguriert haben, können Sie hier auch pro Variablenliste mehrere Verbindungen konfigurieren.

> Das Fenster [Eigenschaften] erweitert sich auf folgendes Bild:



Die Optionen haben dabei folgende Bedeutungen:

Globale Variablenliste: Netzwerktyp

10055

Als Netzwerktyp können Sie einen der bei den Zielsystemeinstellungen angegebenen Netzwerknamen angeben.
Wenn Sie daneben auf die Schaltfläche [Einstellungen] klicken, können Sie die CAN-Schnittstelle wählen:

1. CAN-Schnittstelle: Wert = 0
2. CAN-Schnittstelle: Wert = 1

usw.

Globale Variablenliste: Variablen packen

10056

Wenn diese Option mit [**v**] aktiviert ist, werden die Variablen nach Möglichkeit in einer Übertragungseinheit zusammengefasst. Bei CAN ist eine Übertragungseinheit 8 Bytes groß.

Passen nicht alle Variablen der Liste in eine Übertragungseinheit, dann werden für diese Liste automatisch mehrere Übertragungseinheiten gebildet.

Ist die Option nicht aktiviert, kommt jede Variable in eine eigene Übertragungseinheit.

Wenn [Übertragung bei Änderung] konfiguriert ist, wird für jede Übertragungseinheit getrennt geprüft, ob sie geändert ist und gesendet werden muss.

Globale Variablenliste: Variablenlistenkennung (COB-ID)

10057

Der Basis-Identifier wird als eindeutige Kennung benutzt, um Variablenlisten verschiedener Projekte auszutauschen. Variablenlisten mit gleichem Basis-Identifier werden ausgetauscht. Es ist darauf zu achten, dass die Definitionen der Variablenlisten mit gleichem Basis-Identifier in den verschiedenen Projekten übereinstimmen.

HINWEIS

Der Basis-Identifier wird in CAN-Netzwerken direkt als COB-ID der CAN-Nachrichten benutzt. Es gibt keine Überprüfung, ob der Identifier auch in der übrigen CAN-Konfiguration benutzt wird.

Damit die Daten korrekt zwischen zwei Steuerungen ausgetauscht werden, müssen die globalen Variablenlisten in den beiden Projekten übereinstimmen. Sie können das Feature [Dateiverknüpfung] benutzen, um dies sicherzustellen. Ein Projekt kann die Variablenlisten-Datei vor dem Übersetzen exportieren. Die anderen Projekte sollten diese Datei vor dem Übersetzen importieren.

Neben einfachen Datentypen kann eine Variablenliste auch Strukturen und Arrays enthalten. Die Elemente dieser zusammengesetzten Datentypen werden einzeln versendet.

Es dürfen keine Strings über Netzwerkvariablen verschickt werden, da es sonst zu einem Laufzeitfehler kommt und der Watchdog aktiviert wird.

Wenn eine Variablenliste größer ist als ein PDO des entsprechenden Netzwerks, dann werden die Daten auf mehrere PDOs aufgeteilt. Es kann darum nicht zugesichert werden, dass alle Daten der Variablenliste in einem Zyklus empfangen werden. Teile der Variablenliste können in verschiedenen Zyklen empfangen werden. Dies ist auch für Variablen mit Struktur- und Array-Typen möglich.

© ifm electronic gmbh | www.ifm.com

Globale Variablenliste: Prüfsumme übertragen

10058

Diese Option wird nicht unterstützt.

Globale Variablenliste: Bestätigter Transfer

10059

Diese Option wird nicht unterstützt.

Globale Variablenliste: Lesen

10060

Es werden die Variablenwerte von einer (oder mehreren) Steuerungen gelesen.

Globale Variablenliste: Schreiben

10061

Die Variablen dieser Liste werden zu anderen Steuerungen gesendet.

HINWEIS

Sie sollten für jede Variablenliste nur eine dieser Möglichkeiten auswählen, also entweder nur lesen oder nur schreiben.

Wollen Sie verschiedene Variablen eines Projekts lesen und schreiben, so verwenden Sie bitte mehrere Variablenlisten (eine zum Lesen, eine zum Schreiben).

Für die Kommunikation zwischen 2 Teilnehmern sollten Sie die Variablenliste von einer Steuerung auf die andere kopieren, um die gleiche Datenstruktur zu erhalten.

Zwecks besserer Übersichtlichkeit sollten Ihre Variablenlisten jeweils nur für ein Teilnehmerpaar gelten. Es ist nicht sinnvoll, dieselbe Liste für alle Teilnehmer zu verwenden.

Globale Variablenliste: Zyklische Übertragung

10062

Nur gültig, wenn [Schreiben] aktiviert. Die Werte werden im angegebenen [Intervall] gesendet, unabhängig davon, ob sie sich geändert haben.

Globale Variablenliste: Übertragung bei Änderung

10063

Die Variablenwerte werden nur gesendet, wenn sich einer der Werte geändert hat. Mit [Mindestabstand] (Wert > 0) kann eine Mindestzeit zwischen den Nachrichtenpaketen festgelegt werden.

Globale Variablenliste: Ereignisgesteuerte Übertragung

10064

Wenn diese Option gewählt ist, wird die CAN-Nachricht nur dann übertragen, wenn die angegebene binäre [Variable] auf TRUE gesetzt wird. Diese Variable kann nicht über die Eingabehilfe aus der Liste der definierten Variablen gewählt werden.

© ifm electronic gmbh | www.ifm.com

Besonderheiten bei Netzwerkvariablen

1992

Gerät	Beschreibung
ClassicController: CR0020, CR0505 ExtendedController: CR0200 SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506	Netzwerkvariablen werden nur auf CAN-Schnittstelle 1 (Wert = 0 eintragen) unterstützt. **CANopen-Master** Sende- und Empfangslisten werden direkt verarbeitet. Sie brauchen nur die oben beschriebenen Einstellungen vornehmen. **CANopen-Slave** Sendelisten werden direkt verarbeitet. Für Empfangslisten müssen Sie zusätzlich noch den Bereich der Identifier im Objektverzeichnis auf Empfangs-PDOs mappen. Es ist ausreichend, wenn Sie nur zwei Empfangs-PDOs anlegen und dem ersten Objekt den ersten Identifier und dem zweiten Objekt den letzen Identifier zuweisen. Werden die Netzwerkvariablen nur auf einem Identifier übertragen, müssen Sie nur ein Empfangs-PDO mit diesem Identifier anlegen. Bitte beachten Sie, dass die Identifier der Netzwerkvariablen und der Empfangs-PDOs als **dezimale** Werte eingegeben werden müssen.
ClassicController: CR0032, CR0033 ExtendedController: CR0232, CR0233	Netzwerkvariablen werden auf allen CAN-Schnittstellen unterstützt. (Alle anderen Angaben wie oben)
BasicController: CR0403	Netzwerkvariablen werden auf allen CAN-Schnittstellen unterstützt. (Alle anderen Angaben wie oben)
BasicDisplay: CR0451	Es steht nur eine CAN-Schnittstelle zur Verfügung (Wert = 0 eintragen). (Alle anderen Angaben wie oben)
PDM360: CR1050, CR1051 PDM360compact: CR1052, CR1053, CR1055, CR1056 (nur CAN1 verfügbar)	Netzwerkvariablen werden auf den CAN-Schnittstellen 1 (Wert = 0) und 2 (Wert = 1) unterstützt. **CANopen-Master** Sende- und Empfangslisten werden direkt verarbeitet. Sie brauchen nur die oben beschriebenen Einstellungen vornehmen. **CANopen-Slave** Sende- und Empfangslisten werden direkt verarbeitet. Sie brauchen nur die oben beschriebenen Einstellungen vornehmen. Wird [Netzvariablen unterstützen] im PDM360 oder PDM360compact angewählt, müssen Sie mindestens eine Variable in der Globalen Varablenliste anlegen und diese einmalig im Applikations-Programm aufgerufen. Andernfalls wird die folgende Fehlermeldung bei der Programmübersetzung generiert: Fehler 4601: Netzwerkvariablen 'CAN' : Es ist keine zyklische oder freilaufende Task zum Netwerkvariablenaustausch vorhanden.
PDM360smart: CR1070, CR1071	Es steht nur eine CAN-Schnittstelle zur Verfügung (Wert = 0 eintragen). **CANopen-Master** Sende- und Empfangslisten werden direkt verarbeitet. Sie brauchen nur die oben beschriebenen Einstellungen vornehmen. **CANopen-Slave** Sendelisten werden direkt verarbeitet. Für Empfangslisten müssen Sie zusätzlich noch den Bereich der Identifier im Objektverzeichnis auf Empfangs-PDOs mappen. Es ist ausreichend, wenn Sie nur zwei Empfangs-PDOs anlegen und dem ersten Objekt den ersten Identifier und dem zweiten Objekt den letzen Identifier zuweisen. Werden die Netzwerkvariablen nur auf einem Identifier übertragen, müssen Sie nur ein Empfangs-PDO mit diesem Identifier anlegen. Bitte beachten Sie, dass die Identifier der Netzwerkvariablen und der Empfangs-PDOs als **dezimale** Werte eingegeben werden müssen.
PDM360NG: CR108n	Netzwerkvariablen werden auf allen CAN-Schnittstellen unterstützt. (Alle anderen Angaben wie oben)

© ifm electronic gmbh | www.ifm.com

8.6.2 ifm-CANopen-SDO-Bausteine

Inhalt

CANOPEN_ENABLE 164
CANOPEN_SDOREAD 166
CANOPEN_SDOREADMULTI 168
CANOPEN_SDOWRITE 170
CANOPEN_SDOWRITEMULTI 172

7783

ifm stellt hier dem Applikations-Programmierer eine Reihe von leistungsfähigen Bausteinen zur Verfügung.

© ifm electronic gmbh | www.ifm.com

CANOPEN_ENABLE

7785

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7787

CANOPEN_ENABLE erlaubt es, den CANopen-Master ein- und auszuschalten.

► Im Applikations-Program für jede CAN-Schnittstelle immer eine eigene Instanz des FBs ***CANOPEN_ENABLE*** (→ Seite 164) anlegen!

Zur Vermeidung von Guarding- oder Heartbeat-Fehlern müssen zuvor die Knoten durch eine geeignete Sequenz "heruntergefahren" werden.

Wird der Master nach einem Stopp wieder gestartet, so müssen auch alle angeschlossenen Knoten wieder initialisiert werden.

Ohne CANOPEN_ENABLE wird der CANopen-Master automatisch gestartet, sofern dies in der Konfiguration gewählt wurde.

Die konfigurierte Baud-Rate wird nur übernommen, wenn zuvor nicht ***CAN_ENABLE*** (→ Seite 67) aufgerufen wurde.

Parameter der Eingänge

7788

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Der CANopen-Master startet entsprechend den Konfigurations-Einstellungen TRUE: Der CANopen-Master ist aktiv FALSE: Der CANopen-Master wird beendet oder startet nicht automatisch, auch wenn "Automatischer Start" in den Konfigurationsdaten gewählt ist
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Baudrate (Parameter-Nutzung optional)	WORD	Baud-Rate [kBit/s] zulässige Werte = 20, 50, 100, 125, 250, 500, 800, 1 000 voreingestellt = 0 → Einstellung aus der Steuerungskonfiguration verwenden

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7789

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
014	0E	FB ist aktiv – CANopen-Master konfiguriert Slaves und sendet SDOs
015	0F	FB ist aktiv – CANopen-Master wird gestartet
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Master konnte nicht gestartet werden
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_SDOREAD

7791

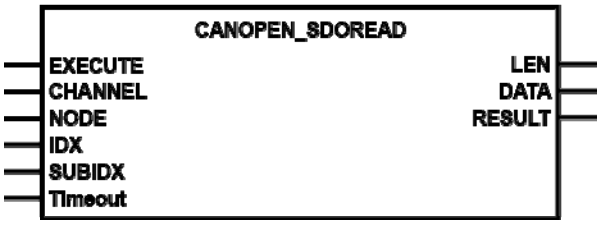
= SDO Read

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7793

CANOPEN_SDOREAD ist ein einfacher Funktionsblock zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

Parameter der Eingänge

7794

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	ID des Knotens zulässige Werte = 01...$7F_{16}$ = 1...127_{10}
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden voreingestellt = 10 ms

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7795

Parameter	Datentyp	Beschreibung
LEN	BYTE	Anzahl der empfangenen Bytes (1...4)
DATA	DWORD	der empfangene Datenwert
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
005	05	FB ist aktiv – noch keine Daten empfangen
032	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 80_{16})
033	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Pufferüberlauf – zu viele Daten-Bytes wurden empfangen

© ifm electronic gmbh | www.ifm.com

CANOPEN_SDOREADMULTI

7806

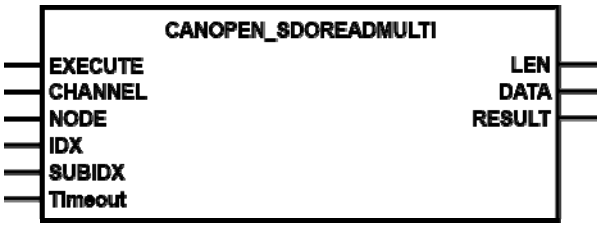
= SDO Read Multi

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7808

CANOPEN_SDOREADMULTI liest den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz. Der COB-ID für den SDO wird nach CANopen-Konvention aus dem übergebenen Node-ID berechnet.

Parameter der Eingänge

7809

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens (1...127) Der COB-ID des SDOs errechnet sich nach CANopen aus Knoten-ID + 600_{16}
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden voreingestellt = 10 ms

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7810

Parameter	Datentyp	Beschreibung
LEN	DWORD	Anzahl der empfangenen Bytes zulässige Werte = 0000 0001...0000 0800_{16} = 1...2 048_{10}
DATA	ARRAY [0...2048] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
005	05	FB ist aktiv – noch keine Daten empfangen
032	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 80_{16})
033	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

© ifm electronic gmbh | www.ifm.com

CANOPEN_SDOWRITE

7825

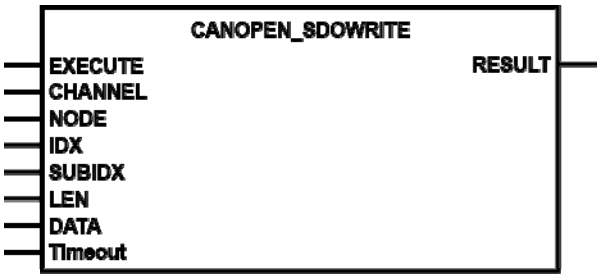
= SDO Write

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7826

CANOPEN_SDOWRITE ist ein einfacher Funktionsblock zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7828

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	ID des Knotens zulässige Werte = $01...7F_{16} = 1...127_{10}$
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis
LEN	BYTE	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = $01...04_{16} = 1...4_{10}$
DATA	ARRAY [0...3] OF BYTE	Datenbereich (1...4 Bytes)
Timeout (Parameter-Nutzung optional)	TIME	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden voreingestellt = 10 ms

Parameter der Ausgänge

7829

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	Funktionsblock ist aktiv
032	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 80_{16})
033	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_SDOWRITEMULTI

7832

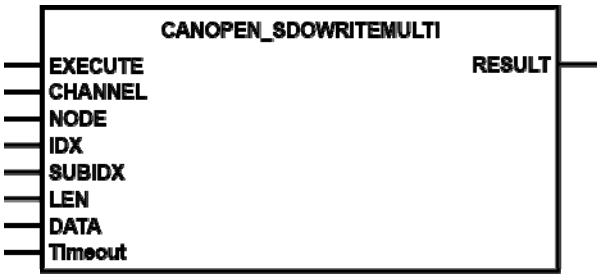
= SDO Write Multi

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7834

CANOPEN_SDOWRITEMULTI schreibt den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz. Der COB-ID für den SDO wird nach CANopen-Konvention aus dem übergebenen Node-ID berechnet.

Parameter der Eingänge

7835

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	ID des Knotens zulässige Werte = 01...$7F_{16}$ = 1...127_{10}
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis
LEN	DWORD	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 0000 0001...0000 0800_{16} = 1...2 048_{10}
DATA	ARRAY [0...2048] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
Timeout (Parameter-Nutzung optional)	TIME	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden voreingestellt = 10 ms

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7836

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	Funktionsblock ist aktiv
032	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 80_{16})
033	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.6.3 CANopen-Netzwerk-Management

Inhalt

Zustandsdiagramm eines CANopen-Knotens ... 175
CANOPEN_NMTSERVICES ... 175
CANOPEN_GETNMTSTATESLAVE ... 177
CANOPEN_SETSTATE ... 178
CANOPEN_GETSTATE ... 181
CANOPEN_SETSYNCSTATE ... 184
CANOPEN_GETSYNCSTATE ... 186
CANOPEN_GETBUFFERFLAGS ... 188
CANOPEN_GETGUARDHBERRLIST ... 190
CANOPEN_GETGUARDHBSTATSLV ... 192
CANOPEN_GETODCHANGEDFLAG ... 194
CANOPEN_READOBJECTDICT ... 195
CANOPEN_WRITEOBJECTDICT ... 196

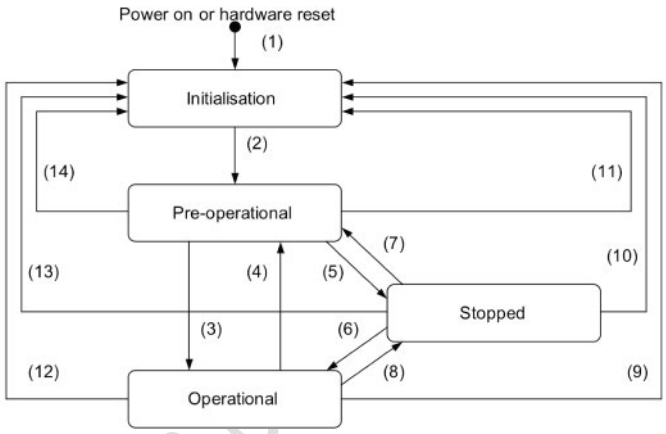
7839

Die folgenden Funktionsbausteine realisieren die Network-Management-Services nach CAN

Zustandsdiagramm eines CANopen-Knotens

7840

Das Netzwerk-Management überprüft anhand der internen Listen, ob ein angeforderter Zustandswechsel erlaubt ist. Wenn der Übergang nicht erlaubt ist, wird das Kommando nicht gesendet, sondern eine Fehlermeldung ausgegeben.



Grafik: Zustandsübergänge unter CANopen

Erlaubte Übergänge:

(1) Zustand wird bei Power On automatisch erreicht.

(2) interne Initialisierung ist beendet – Knoten geht automatisch nach PRE-OPERATIONAL

(3) NMT Service "Start Remote Node Indication"

(4) + (7) NMT Service "Enter PRE-OPERATIONAL Indication"

(5) + (8) NMT Service "Stop Remote Node Indication"

(6) NMT Service "Start Remote Node Indication"

(9)...(11) NMT Service "Reset Node Indication"

(12)...(14) NMT Service "Reset Communication Indication"

Der Funktionsbaustein überprüft anschließend, ob die Statusänderung an dem Knoten erfolgreich war, soweit dies vom Protokoll möglich ist.

Der CANopen-Zustand der Steuerung wird im CANopen-Stack behandelt. Läuft die Steuerung im Slave-Mode und empfängt von ihrem Master NMT-Kommandos, so werden diese im CANopen-Stack behandelt und verwaltet. Die Zustände der betreffenden Flags und ihr Zurücksetzen müssen daher nicht mehr im Applikations-Programm bedient werden.

CANOPEN_NMTSERVICES

7843

= Network Management Services

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7844

CANOPEN_NMTSERVICES löst abhängig von seinen NMT-Kommando-Eingängen ein NMT-Kommando oder die Initialisierung eines Knotens aus.

NMT = **N**etwork-**M**anagemen**T**

Der Funktionsblock aktualisiert den internen Knotenstatus. Sollte ein Zustandsübergang nach CANopen (→ ***Zustandsdiagramm eines CANopen-Knotens*** (→ Seite 175)) nicht erlaubt sein, so wird das Kommando nicht ausgeführt.

Ein CANopen-Device kann mit Hilfe des FB seinen CANopen-Status selbständig ändern: Preoperational ⇦⇨ Operational

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7847

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODEID	BYTE	Node-ID = ID des Knotens ($0...127_{10}$) NODEID = 0: Kommando gilt für alle am Netzwerk angeschlossenen Knoten NODEID = Node-ID des Geräts: Kommando gilt für das Gerät selbst
NMTSERVICE	BYTE	0 = Init Node (außer Master) 1 = Enter PRE-OPERATIONAL 2 = Start Node 3 = Reset Node 4 = Reset Kommunikation 5 = Stop Node
Timeout (Parameter-Nutzung optional)	TIME	Wartezeit des FB auf die Initialisierung Nach Zeitablauf bricht der FB das Warten ab. Wert = 0 (voreingestellt): Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7848

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist aktiv
035	23	mindestens 1 SDO der Konfiguration war nicht erfolgreich
036	24	Knoten war bereits initialisiert
037	25	zur Initialisierung war Knoten nicht im Modus PRE-OPERATIONAL
043	2B	Master / Slave ist nicht initialisiert
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETNMTSTATESLAVE

7851

= Get Network Management State Slave

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7853

► Baustein nur verwenden, wenn das Gerät als CANopen-Slave betrieben wird!

An die Applikation wird mit CANOPEN_GETNMTSTATESLAVE nur noch der Betriebszustand nach CANopen gemeldet sowie eine Fehlermeldung, falls ein ungültiger Zustandsüberangefordert wurde.

Parameter der Eingänge

7854

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7855

Parameter	Datentyp	Beschreibung
NMTSTATE	BYTE	Netzwerk-Betriebszustand des Knotens 0 = INIT 1 = OPERATIONAL 2 = PRE-OPERATIONAL 3 = STOPPED
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_SETSTATE

7858

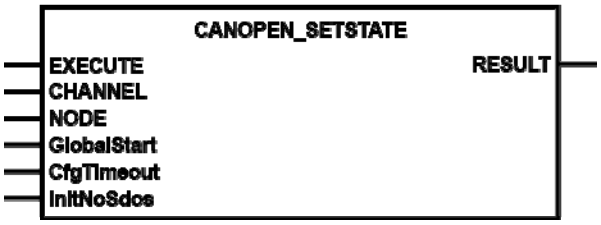
= Set State

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7860

Mit CANOPEN_SETSTATE können Parameter des Masters, eines Slave-Devices oder eines Knotens im Netz gesetzt werden.

Die Behandlung des NMT-Zustands von Master, Knoten oder Device erfolgt im CAN-Stack oder über die Kommandos des FB ***CANOPEN_NMTSERVICES*** (→ Seite 175). Dabei werden gleichzeitig auch Zulässigkeitsprüfungen durchgeführt. Aus Konsistenzgründen sind deshalb hier für diesen Zweck keine Eingänge vorgesehen.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7861

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) **Gerät als CANopen-Master:** Wert = 0: Die Änderungen beziehen sich nur auf das Gerät selbst. Wert nicht 0: Node-ID eines Knotens am Netzwerk, dessen Parameter verändert werden sollen. Nur für diesen Knoten (nicht für dias Gerät) werden die angelegten Einstellungen übernommen. **Gerät als CANopen-Slave:** Im Slave-Mode kann über diesen Eingang der Node-ID des Slave gesetzt werden. Wert = 0: keine Aktion Wert nicht 0: Diesen Wert übernimmt der FB als neuen Node-ID des Geräts.
GlobalStart (Parameter-Nutzung optional)	BOOL	Voraussetzung: FB muss unmittelbar nach dem Start des IEC-Programms aufgerufen werden. Diese Einstellung überschreibt die Einstellung aus der Konfiguration. TRUE (voreingestellt): Bei der Netzwerk-Initialisierung werden alle Teilnehmer gleichzeitig gestartet. FALSE: Bei der Netzwerk-Initialisierung werden alle Teilnehmer nacheinander gestartet.
CfgTimeout (Parameter-Nutzung optional)	TIME	Konfigurations-Timeout für einen Knoten setzen: Wert = 0 (voreingestellt): keine Aktion – Konfigurationsdaten behalten Wert nicht 0: Daten aus der Konfiguration mit dem neuen Wert überschreiben.
InitNoSdos (Parameter-Nutzung optional)	BOOL	TRUE: An den in NODE angegebenen Knoten werden beim Initialisieren keine Konfigurationsdaten gesendet. FALSE (voreingestellt): An den in NODE angegebenen Knoten werden beim Initialisieren die konfigurierten SDOs gesendet.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7862

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETSTATE

7865

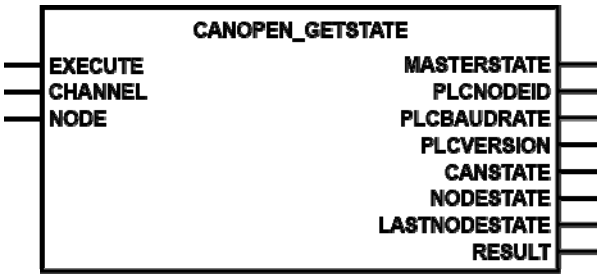
= Get State

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7867

Mit CANOPEN_GETSTATE können Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz abgefragt werden.

Parameter der Eingänge

7868

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) **Gerät als CANopen-Master:** Wert = 0: Nur die Statusinformationen des Geräts selbst werden an den Ausgängen zurückgeliefert. Die Ausgänge mit Informationen zu den Knoten sind ungültig. Wert nicht 0: Node-ID eines Knotens im Netzwerk. Für diesen sowie für das Gerät werden an den Ausgängen die Zustände zurückgeliefert. **Gerät als CANopen-Slave:** Wert = 0 (voreingestellt): Die Statusinformationen des Slave werden an den Ausgängen zurückgeliefert. Wert nicht 0: keine Aktion

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7869

Parameter	Datentyp	Beschreibung
MASTERSTATE	BYTE	Master State = interner Zustand des Masters:
		0_{10} 00_{16} Master läuft gerade hoch
		4_{10} 04_{16} Konfiguration der Knoten läuft
		5_{10} 05_{16} normaler Betriebszustand des Masters
		255_{10} FF_{16} wenn PLC als Slave läuft
PLCNODEID	BYTE	PLC Node-ID = Node-ID der Steuerung, auf der das Programm ausgeführt wird Wert = $0...127_{10} = 0...7F_{16}$
PLCBAUDRATE	DWORD	Ausgabe der Baudrate der SPS
PLCVERSION	DWORD	Version der Steuerung
CANSTATE	BYTE	CAN State = Status des CANopen-Netzwerks **Gerät als Master betrieben:** **Node-ID = 0 (Gerät selbst):** $00_{10} = 00_{16}$ = OK $128_{10} = 80_{16}$ = Busoff **Node-ID ≠ 0 (Knoten):** $00_{10} = 00_{16}$ = OK $01_{10} = 01_{16}$ = Guard- oder Heartbeat-Fehler an Knoten $128_{10} = 80_{16}$ = Busoff **Gerät als Slave betrieben:** $00_{10} = 00_{16}$ = OK $01_{10} = 01_{16}$ = Guard- oder Heartbeat-Fehler $128_{10} = 80_{16}$ = Busoff
NODESTATE	BYTE	Node State = interner Knotenstatus eines Slaves aus Sicht des Masters. Der Knoten wird durch den Eingang NODEID bezeichnet.
		-1_{10} FF_{16} Rücksetzen nach ResetNode
		1_{10} 01_{16} Warten auf BOOTUP
		2_{10} 02_{16} Nach Empfang BOOTUP-Nachricht
		3_{10} 03_{16} noch nicht konfiguriert: PREPARED
		4_{10} 04_{16} nach der Konfiguration mit SDOs: PRE-OPERATIONAL
		5_{10} 05_{16} nach dem Starten des Knoten: OPERATIONAL
		97_{10} 61_{16} optionaler Knoten
		98_{10} 62_{16} anderer Gerätetyp als in 1000_{16} konfiguriert
		99_{10} 63_{16} Node-Guarding
LASTNODESTATE	BYTE	Last Node State = letzter Status des Knotens Knotenstatus nach CANopen (mit diesen Werten wird der Status auch in den entsprechenden Nachrichten vom Knoten her codiert).
		0_{10} 00_{16} BOOTUP
		4_{10} 04_{16} PREPARED
		5_{10} 05_{16} OPERATIONAL
		127_{10} $7F_{16}$ PRE-OPERATIONAL
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

© ifm electronic gmbh | www.ifm.com

Mögliche Ergebnisse für RESULT:

Wert dez | hex		Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_SETSYNCSTATE

7883

= Set Sync State

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7884

Mit CANOPEN_SETSYNCSTATE wird die SYNC-Funktionalität ein- und ausgeschaltet.

Parameter der Eingänge

7886

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert: Im **Master-Betrieb** werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im **Slave-Betrieb** werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7887

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist noch nicht ausgeführt
038	26	SYNC konnte nicht aktiviert werden
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETSYNCSTATE

7871

= Get Sync State

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7872

CANOPEN_GETSYNCSTATE liest...
- die Einstellung der SYNC-Funktionalität (aktiv / deaktiv),
- den Fehlerzustand der SYNC-Funktionalität (SyncError).

Wenn die PLC als CANopen-Slave läuft, wird über diesen FB signalisiert, ob SYNC-Signale ausbleiben oder ob sie regelmäßig kommen.

Die Bearbeitung von synchronen PDOs usw. läuft im CAN-Stack. CANOPEN_GETSYNCSTATE liefert jedoch den Fehlerzustand, so dass das Applikations-Programm darauf entsprechend reagieren kann.

Parameter der Eingänge

7874

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7875

Parameter	Datentyp	Beschreibung
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert: Im **Master-Betrieb** werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im **Slave-Betrieb** werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv
SYNCERROR	BYTE	(Sync-Error) SYNC-Fehlermeldung 0 = kein Fehler >0 = SYNC-Error (Slave-Betrieb)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETBUFFERFLAGS

7890

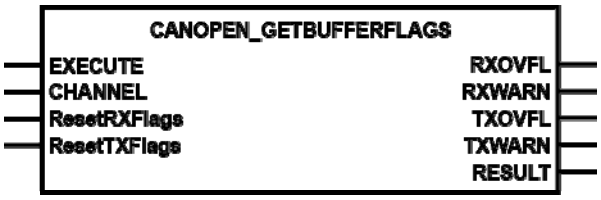
= Get Buffer-Flags

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7892

CANOPEN_GETBUFFERFLAGS liefert Informationen zu den Buffer-Flags.

Über optionale Eingänge können die Flags zurückgesetzt werden.

Der Funktionsbaustein liefert den Zustand der Overflow-Flags zurück.

Parameter der Eingänge

7893

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetRXFlags (Parameter-Nutzung optional)	BOOL	TRUE: Flag-Status wird am Ausgang ausgegeben und anschließend zurückgesetzt. FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)
ResetTXFlags (Parameter-Nutzung optional)	BOOL	TRUE: Flag-Status wird am Ausgang ausgegeben und anschließend zurückgesetzt. FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7894

Parameter	Datentyp	Beschreibung
RXOVFL	BOOL	Zustand des RX-Overflow-Flags TRUE: Überlauf im Empfangspuffer FALSE: Kein Überlauf im Empfangspuffer
RXWARN	BOOL	Zustand des RX-Overflow-Warning-Flags TRUE: Füllstand im Empfangspuffer ist kritisch FALSE: Füllstand im Empfangspuffer ist unkritisch
TXOVFL	BOOL	Zustand des TX-Overflow-Flags TRUE: Überlauf im Sendepuffer FALSE: Kein Überlauf im Sendepuffer
TXWARN	BOOL	Zustand des TX-Overflow-Warning-Flags TRUE: Füllstand im Sendepuffer ist kritisch FALSE: Füllstand im Sendepuffer ist unkritisch
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh · www.ifm.com

CANOPEN_GETGUARDHBERRLIST

7896

= Get Guard and Heartbeat Error-List

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7898

CANOPEN_GETGUARDHBERRLIST listet in einem Array alle Knoten auf, für die der Master einen Fehler erkannt hat:
- Guarding-Fehler
- Heartbeat-Fehler

Parameter der Eingänge

7899

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetList (Parameter-Nutzung optional)	BOOL	Fehlerliste zurücksetzen TRUE: Die Fehlerliste sowie die Anzahl der fehlerhaften Knoten werden am Ausgang ausgegeben und anschließend zurückgesetzt FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7900

Parameter	Datentyp	Beschreibung
N_NODES	WORD	Wert = 0: Kein Knoten hat einen Guarding- oder Heartbeat-Fehler. Wert nicht 0: Anzahl der Knoten mit Heartbeat- oder Guarding-Fehlern
NODEID	ARRAY [0...MAXGUARDERROR] OF BYTE	Ab Index 0 werden die Knoten-IDs eingetragen, bei denen ein Heartoder Guarding-Fehler vorliegt. Der jüngste Eintrag steht im Index 0. MAXGUARDERROR ist abhängig vom Gerät → Kapitel ***Leistungsgrenzen der Geräte (CANopen)*** (→ Seite 131)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETGUARDHBSTATSLV

7902

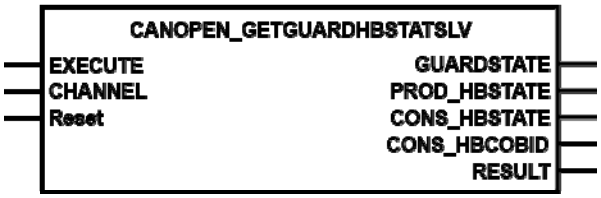
= Get Guard and Heartbeat State Slave

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7904

CANOPEN_GETGUARDANDHBSTATESLAVE meldet der Steuerung im Slave-Betrieb folgende Zustände:
- Node-Guarding-Überwachung
- Heartbeat-Überwachung

Dabei kann die Steuerung Heartbeat-Producer und Heartbeat-Consumer sein.

Parameter der Eingänge

7905

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset (Parameter-Nutzung optional)	BOOL	TRUE: Die aktuellen Zustände weran den Ausgängen ausgegeben und anschließend wird auf "Kein Fehler" zurückgesetzt. FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7906

Parameter	Datentyp	Beschreibung
GUARDSTATE	BYTE	Status des Nodeguarding: $00_{10} = 00_{16}$ = kein Fehler (oder: inaktiv) $01_{10} = 01_{16}$ = Zeitüberschreitung (Konfiguration) $127_{10} = 7F_{16}$ = noch keine Guarding-Nachricht empfangen
PROD_HBSTATE	BYTE	Steuerung als Heartbeat-Producer: $00_{10} = 00_{16}$ = inaktiv $01_{10} = 01_{16}$ = aktiv
CONS_HBSTATE	BYTE	Steuerung als Heartbeat-Consumer: $00_{10} = 00_{16}$ = kein Fehler $01_{10} = 01_{16}$ = Zeitüberschreitung (Konfiguration) $127_{10} = 7F_{16}$ = noch keine Heartbeat-Nachricht empfangen
CONS_HBCOBID	BYTE	COB-ID der Heartbeat-Nachricht, auf die der Consumer-Heartbeat der Steuerung hört (Konfiguration)
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETODCHANGEDFLAG

7927

= Get Object Directory Changed Flag

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7928

CANOPEN_GETODCHANGEDFLAG meldet wenn bei einem bestimmten Objektverzeichnis-Eintrag der Wert geändert wurde.

Parameter der Eingänge

7930

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis

Parameter der Ausgänge

7931

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_READOBJECTDICT

7933

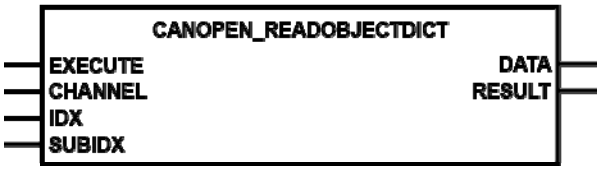
= Read Object Directory

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7935

CANOPEN_READOBJECTDICT liest Konfigurationsdaten aus dem Objektverzeichnis der Steuerung zur Verwendung im Applikations-Programm.

Parameter der Eingänge

7936

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis

Parameter der Ausgänge

7937

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist noch nicht ausgeführt
040	28	Objektverzeichnis-Eintrag ist ungültig
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_WRITEOBJECTDICT

7940

= Write Object Directory

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7942

CANOPEN_WRITEOBJECTDICT schreibt Konfigurationsdaten in das Objektverzeichnis der Steuerung.

ACHTUNG

Wichtige Systemeinstellungen können hierbei verfälscht werden, z.B.:
- Guarding-Zeiten
- Heartbeat-Zeiten

► Eingabe-Parameter sorgfältig prüfen!

Parameter der Eingänge

7943

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex im Objektverzeichnis
DATA	DWORD	Parameter-Wert

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7945

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	Funktionsblock ist noch nicht ausgeführt
040	28	Objektverzeichnis-Eintrag ist ungültig
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

8.7 CAN-Fehler und Fehlerbehandlung

Inhalt

Aufbau einer EMCY-Nachricht 199
Übersicht CANopen Error-Codes 201
CANopen-Bausteine zur Fehlerbehandlung 204

7948

8.7.1 Aufbau einer EMCY-Nachricht

8045

Die Signalisierung von Fehlerzuständen erfolgt unter CANopen über einen sehr einfachen, standardisierten Mechanismus. Jedes Auftreten eines Fehlers bei einem CANopen-Gerät wird über eine spezielle Nachricht signalisiert, die den Fehler genauer beschreibt.

Verschwindet ein Fehler oder seine Ursache nach einer bestimmten Zeit wieder, wird dieses Ereignis ebenfalls über die EMCY-Nachricht signalisiert.

> Die zuletzt aufgetretenen Fehler werden im Objektverzeichnis (Objekt 1003_{16}) abgelegt und können mit ***CANOPEN_SDOREAD*** (→ Seite 166) ausgelesen werden.

> Die aktuelle Fehlersituation steht im Error-Register (Objekt 1001_{16}).

Man unterscheidet folgende Fehler:

8046

Kommunikationsfehler

- Der CAN-Controller signalisiert CAN-Fehler.
  (Das gehäufte Auftreten ist ein Indiz für physikalische Probleme. Diese Fehler können einen erheblichen Einfluss auf das Übertragungsverhalten und damit auf den Datendurchsatz eines Netzwerks haben.)
- Life-Guarding- oder Heartbeat-Fehler

Anwendungsfehler

- Kurzschluss oder Leiterbruch
- Temperatur zu hoch

Aufbau einer Fehlernachricht

8047

Eine Fehlernachricht (EMCY Message) hat folgenden Aufbau:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EMCY-Fehlercode, wie im Objekt 1003_{16} eingetragen		Objekt 1001_{16}	Herstellerspezifische Informationen				

Identifier

8048

Der Identifier für die Fehlernachricht besteht aus der Summe folgender Elemente:

EMCY-Default-Identifier 128 (80_{16})
+
Node-ID

ifm electronic gmbh | www.ifm.com

EMCY-Fehlercode

8049

Er gibt detailliert Auskunft darüber, welcher Fehler aufgetreten ist. Eine Liste möglicher Fehlercodes ist bereits im Kommunikationsprofil definiert. Fehlercodes, die nur für eine bestimmte Geräteklasse gültig sind, werden im jeweiligen Geräteprofil dieser Geräteklasse festgelegt.

Objekt 0x1003 (Error Field)

8050

Das Objekt 1003_{16} stellt den Fehlerspeicher eines Gerätes dar. Die Subindizes enthalten die zuletzt aufgetretenen Fehler, die ein Fehler-Telegramm ausgelöst haben.

Tritt ein neuer Fehler auf, dann wird sein EMCY-Fehlercode immer im Subindex 1_{16} gespeichert. Alle anderen, älteren Fehler werden im Fehlerspeicher um einen Platz nach hinten geschoben, also der Subindex um 1 erhöht. Falls alle unterstützten Subindizes belegt sind, wird der älteste Fehler gelöscht. Der Subindex 0_{16} wird auf die Anzahl der gespeicherten Fehler erhöht. Nachdem alle Fehler behoben sind, wird in das Fehlerfeld des Subindex 1_{16} der Wert "0" geschrieben.

Um den Fehlerspeicher zu löschen, kann der Subindex 0_{16} mit dem Wert "0" beschrieben werden. Andere Werte dürfen nicht eingetragen werden.

Gerätefehler signalisieren

8051

Wie beschrieben, werden EMCY-Nachrichten versendet, wenn Fehler in einem Gerät auftreten.

> Im Unterschied zu frei programmierbaren Geräten, werden beispielsweise von dezentralen Ein-/Ausgangsmodulen (z.B. CompactModule CR2033) Fehlermeldungen automatisch verschickt. Entsprechende Fehler-Codes → jeweiliges Gerätehandbuch.

> Die programmierbaren Geräte erzeugen nur für Guarding- und Heartbeat-Fehler automatisch eine EMCY-Nachricht.
Übersicht von EMCY-Fehlercodes für alle mit CoDeSys programmierbaren ecomatmobile-Geräte → Kapitel ***Übersicht CANopen Error-Codes*** (→ Seite 201).

Sollen zusätzlich noch applikations-spezifische Fehler durch das Applikations-Programm verschickt werden, wird ***CANOPEN_SENDEMCYMESSAGE*** (→ Seite 205) eingesetzt.

© ifm electronic gmbh | www.ifm.com

8.7.2 Übersicht CANopen Error-Codes

8545

Error Code (hex)	Meaning / Bedeutung
00xx	Reset or no Error (Fehler rücksetzen / kein Fehler)
10xx	Generic Error (allgemeiner Fehler)
20xx	Current (Stromfehler)
21xx	Current, device input side (Stromfehler, eingangsseitig)
22xx	Current inside the device (Stromfehler im Geräteinnern)
23xx	Current, device output side (Stromfehler, ausgangsseitig)
30xx	Voltage (Spannungsfehler)
31xx	Mains Voltage
32xx	Voltage inside the device (Spannungsfehler im Geräteinnern)
33xx	Output Voltage (Spannungsfehler, ausgangsseitig)
40xx	Temperature (Temperaturfehler)
41xx	Ambient Temperature (Umgebungstemperaturfehler)
42xx	Device Temperature (Gerätetemperaturfehler)
50xx	Device Hardware (Geräte-Hardware-Fehler)
60xx	Device Software (Geräte-Software-Fehler)
61xx	Internal Software (Firmware-Fehler)
62xx	User Software (Applications-Software)
63xx	Data Set (Daten-/Parameterfehler)
70xx	Additional Modules (zusätzliche Module)
80xx	Monitoring (Überwachung)
81xx	Communication (Kommunikation)
8110	CAN Overrun-objects lost (CAN Überlauf-Datenverlust)
8120	CAN in Error Passiv Mode (CAN im Modus "fehlerpassiv")
8130	Life Guard Error or Heartbeat Error (Guarding-Fehler oder Heartbeat-Fehler)
8140	Recovered from Bus off (Bus-Off zurückgesetzt)
8150	Transmit COB-ID collision (Senden "Kollision des COB-ID")
82xx	Protocol Error (Protokollfehler)
8210	PDO not procedded due to length error (PDO nicht verarbeitet, fehlerhafte Längenangabe)
8220	PDO length exceeded (PDO Längenfehler, ausgangsseitig)
90xx	External Error (Externer Fehler)
F0xx	Additional Functions (zusätzliche Funktionen)
FFxx	Device specific (gerätespezifisch)

©ifm electronic gmbh | www.ifm.com

Objekt 0x1001 (Error-Register)

8547

Dieses Objekt spiegelt den allgemeinen Fehlerzustand eines CANopen-Gerätes wider. Das Gerät ist dann als fehlerfrei anzusehen, wenn das Objekt 1001_{16} keinen Fehler mehr signalisiert.

Bit	Meaning (Bedeutung)
0	Generic Error (allgemeiner Fehler)
1	Current (Stromfehler)
2	Voltage (Spannungsfehler)
3	Temperature (Temperaturfehler)
4	Communication Error (Kommunikationsfehler)
5	Device Profile specific (Geräteprofil spezifisch)
6	Reserved – always 0 (reserviert – immer 0)
7	manufacturer specific (herstellerspezifisch)

Für eine Fehlermeldung können mehrere Bits im Error-Register gleichzeitig gesetzt sein.

Beispiel: CR2033, Meldung "Leitungsbruch" an Kanal 2 (→ Installationsanleitung des Geräts):

COB-ID	DLC	Byte 0	Byte 1	Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
80_{16} + Node-ID		**00**	**FF**	**81**	**10**	**00**	00	00	00

Error-Code = $FF00_{16}$

Error-Register = 81_{16} = $1000\ 0001_2$, besteht also aus folgenden Fehlern:
- Generic Error (allgemeiner Fehler)
- manufacturer specific (herstellerspezifisch)

Betroffener Kanal = 0010_{16} = $0000\ 0000\ 0001\ 0000_2$ = Kanal 2

© ifm electronic gmbh | www.ifm.com

Herstellerspezifische Informationen

8548

Hier kann ein Gerätehersteller zusätzliche Fehlerinformationen mitteilen. Das Format ist dabei frei wählbar.

Beispiel:

In einem Gerät treten zwei Fehler auf und werden über den Bus gemeldet:

- Kurzschluss der Ausgänge:
  Fehlercode 2300_{16},
  im Objekt 1001_{16} wird der Wert 03_{16} ($0000\ 0011_2$) eingetragen
  (allg. Fehler und Stromfehler)

- CAN-Überlauf:
  Fehlercode 8110_{16},
  im Objekt 1001_{16} wird der Wert 13_{16} ($0001\ 0011_2$) eingetragen
  (allg. Fehler, Stromfehler und Kommunikationsfehler)

>> CAN-Überlauf bearbeitet:
Fehlercode 0000_{16},
im Objekt 1001_{16} wird der Wert 03_{16} ($0000\ 0011_2$) eingetragen
(allg. Fehler, Stromfehler, Kommunikationsfehler zurückgesetzt.)

Nur aus dieser Information kann man entnehmen, dass der Kommunikationsfehler nicht mehr anliegt.

Übersicht CANopen EMCY-Codes (CR040n)

8412

EMCY-Code Objekt 1003_{16}		Objekt 1001_{16}	Hersteller-spezifische Informationen					
Byte 0	1	2	3	4	5	6	7	Beschreibung
00	21	03	I0 LSB	I0 MSB				Leiterbruch Eingänge
08	21	03	I0 LSB	I0 MSB				Kurzschluss Eingänge
10	21	03	I0 LSB	I0 MSB				Überstrom 4…20 mA
00	23	03	Q0 LSB	Q0 MSB				Leitungsbruch Ausgänge
08	23	03	Q0 LSB	Q0 MSB				Kurzschluss Ausgänge
00	42	09						Übertemperatur
00	31	05						Versorgungsspannung
00	33	05						Klemmenspannung VBBo
08	33	05						Ausgangsspannung VBBr
00	80	11						CAN1 Monitoring SYNC-Error (nur Slave)
00	81	11						CAN1 Warngrenze (> 96)
10	81	11						CAN1 Empfangspuffer Überlauf
11	81	11						CAN1 Sendepuffer Überlauf
30	81	11						CAN1 Guard-/Heartbeat-Error (nur Slave)

Im CANopen-Stack ist noch keiner dieser EMCY-Codes fix implementiert. Vorschlag:

► Diese EMCY-Codes mit CANOPEN_SENDEMCYMESSAGE erzeugen.

© ifm electronic gmbh | www.ifm.com

8.7.3 CANopen-Bausteine zur Fehlerbehandlung

Inhalt

CANOPEN_SENDEMCYMESSAGE 205
CANOPEN_GETERRORREGISTER 206
CANOPEN_GETEMCYMESSAGES 208

8522

© ifm electronic gmbh | www.ifm.com

CANOPEN_SENDEMCYMESSAGE

7908

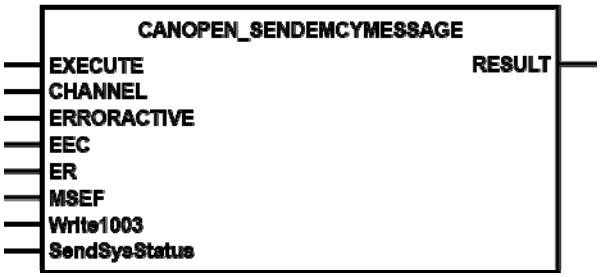
= Send Emergency-Message

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:



Beschreibung

7910

CANOPEN_SENDEMCYMESSAGE versendet eine EMCY-Nachricht. Die Nachricht wird aus den entsprechenden Parametern zusammengebaut und ins Register 1003_{16} eingetragen. Der COB-ID für die Emergency-Nachricht wird aus den Konfigurationsdaten ermittelt.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

7911

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ERRORACTIVE	BOOL	FALSE ⇨ TRUE (Flanke): sendet den anstehenden Fehlercode TRUE ⇨ FALSE (Flanke): Wenn der Fehler NICHT mehr ansteht, wird nach einer Verzögerung von ca. 1 s eine Null-Fehlermeldung gesendet.
EEC	WORD	EEC = Emergency Error Code = Fehlermeldungsnummer
ER (Parameter-Nutzung optional)	BYTE	0 = Wert aus dem Objekt 1001_{16}, Error Register wird verwendet. (voreingestellt)
MSEF	ARRAY [0...4] OF BYTE	MSEF = Manufacturer Specific Error Code = Zusätzlicher Fehler-Code, der vom Hersteller festgelegt wird. Wert kommt aus der Applikation.
Write1003 (Parameter-Nutzung optional)	BOOL	TRUE: Diese EMCY-Nachricht im Objekt 1003_{16} eintragen FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)
SendSysStatus (Parameter-Nutzung optional)	BOOL	Send System-Status TRUE: Der Systemstatus wird überprüft und bei Vorliegen eines Fehlerstatus wird dieser ins Netzwerk übertragen. FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)

Parameter der Ausgänge

7912

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
039	27	kein Objekt 1001_{16} in der Konfiguration
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETERRORREGISTER

7915

= Get Error-Register

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7917

CANOPEN_GETERRORREGISTER liest die Fehler-Register 1001_{16} und 1003_{16} der Steuerung aus.

Parameter der Eingänge

7918

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv, ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset_1001 (Parameter-Nutzung optional)	BOOL	TRUE: Das Fehler-Register 1001_{16} der Steuerung wird zurückgesetzt FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)
Reset_1003 (Parameter-Nutzung optional)	BOOL	TRUE: Das Fehler-Register 1003_{16} der Steuerung wird zurückgesetzt. Die Anzahl der Einträge wird auf 0 gesetzt. FALSE: diese Funktion wird nicht ausgeführt (voreingestellt) Die Einträge bleiben unverändert.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7919

Parameter	Datentyp	Beschreibung
ER	BYTE	Inhalt des Fehler-Registers 1001_{16}
ERROR_FIELD	ARRAY [0...MAXERR] OF DWORD	Inhalt des Error-Registers 1003_{16} Index 0 = Anzahl der gespeicherten Fehler Index 1...MAXERR = gespeicherte Fehler Der jüngste Fehler steht im Index 1 voreingestellt: MAXERR = 5
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

CANOPEN_GETEMCYMESSAGES

7921

= Get Emergency-Messages

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CANopen_NT_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Symbol in CoDeSys:

Beschreibung

7923

CANOPEN_GETEMCYMESSAGES gibt alle Emergency-Nachrichten zurück, die die Steuerung seit dem letzten Löschen der Nachrichten von anderen Knoten am Netz empfangen hat.

Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden. Es werden maximal MAXEMCYMSGS Nachrichten gespeichert. Jede Nachricht enthält dabei als Info, von welchem Knoten sie gesendet wurde. Dabei steht die jüngste Nachricht im Index 0.

Parameter der Eingänge

7924

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird einmalig ausgeführt sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RstList (Parameter-Nutzung optional)	BOOL	TRUE: Liste mit aufgelaufenen CAN-Nachrichten wird am Ausgang ausgegeben und anschließend gelöscht FALSE: diese Funktion wird nicht ausgeführt (voreingestellt)

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

7925

Parameter	Datentyp	Beschreibung
N_MSGS	DWORD	Anzahl der aufgelaufenen Nachrichten Höchster Index bezeichnet die älteste Nachricht
EMCY	ARRAY [0...MAXEMCYMSGS] OF T_EMCY	Emergency-Nachrichten Struktur von T_EMCY: .NODEID – ID des Knotens von dem die Nachricht kam .EEC – Emergency Error Code .ER – Error Register .MSEF – Manufacturer Specific Error Code MAXEMCYMSG = 10
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
008	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh | www.ifm.com

9 Ein-/Ausgangs-Funktionen

Inhalt

Eingangswerte verarbeiten 211
Zählerfunktionen zur Frequenz- und Periodendauermessung 215
Ausgangsfunktionen allgemein 223
PWM-Funktionen 227

1590

Hier zeigen wir Ihnen Funktionen zum Lesen und Bearbeiten der Signale an den Ein- und Ausgängen.

9.1 Eingangswerte verarbeiten

Inhalt

INPUT 212

1602

Hier zeigen wir Ihnen Funktionen zum Lesen und Verarbeiten der analogen oder digitalen Signale am Geräte-Eingang.

HINWEIS

Die in der Steuerungskonfiguration von CoDeSys erscheinenden Rohwerte kommen direkt aus dem ADU. Sie sind noch nicht korrigiert!

Deshalb können in der Steuerungskonfiguration bei gleichen Geräten unterschiedliche Rohwerte erscheinen.

Erst durch die ifm-FBs (z.B. INPUT, INPUT_ANALOG) findet eine Fehlerkorrektur und Normierung statt. Die FBs liefern den korrigierten Wert.

© ifm electronic gmbh | www.ifm.com

9.1.1 INPUT

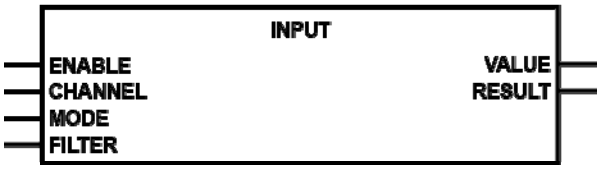
8103

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8104

INPUT ermöglicht die Zustandserfassung an den Eingangskanälen (→ Datenblatt).

Der FB liefert den aktuellen Zustand am gewählten Kanal. Die Messung und der Ausgangswert resultiert aus der über MODE angegebenen Betriebsart:
- Digital-Eingang (lowside) mit/ohne Diagnose
- Digital-Eingang (highside)
- Analog-Eingang 0...20 mA
- Analog-Eingang 0...10 V
- Analog-Eingang 0...32 V
- Analog-Eingang ratiometrische Messung 0...32 V
- Analog-Eingang Widerstandsmessung 0...3600 Ohm

Im laufenden Betrieb sollte die Betriebsart MODE nicht geändert werden.

Die Analogwerte werden normiert ausgegeben.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

8105

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Eingangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
MODE	BYTE	Betriebsart des Eingangskanals:
		$00_{10} = 00_{16}$ Aus
		$03_{10} = 03_{16}$ Spannungseingang 0…10 000 mV
		$06_{10} = 06_{16}$ Spannungseingang ratiometrisch 0...1 000 ‰
		$07_{10} = 07_{16}$ Stromeingang 0…20 000 µA
		$09_{10} = 09_{16}$ Spannungseingang 0…32 000 mV
		$10_{10} = 0A_{16}$ Binäreingang Lowside (voreingestellt)
		$11_{10} = 0B_{16}$ Binäreingang Lowside mit Diagnose (Namur)
		$12_{10} = 0C_{16}$ Binäreingang Highside
		$18_{10} = 12_{16}$ Widerstandseingang 0…3 600 Ω
FILTER	BYTE	Filter für die Messung am Eingang
		0_{10} 00_{16} kein Filter
		1_{10} 01_{16} 120 Hz
		2_{10} 02_{16} 47 Hz
		3_{10} 03_{16} 22 Hz
		4_{10} 04_{16} 10 Hz (empfohlen)
		5_{10} 05_{16} 5 Hz
		6_{10} 06_{16} 2,5 Hz
		7_{10} 07_{16} 1,2 Hz
		8_{10} 08_{16} 0,7 Hz

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8106

Parameter	Datentyp	Beschreibung
VALUE	WORD	aktueller Wert oder Zustand des Eingangskanals
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
136	88	Filtereinstellung ist ungültig
141	8D	Leiterbruch ist aufgetreten
142	8E	Schluss gegen Versorgung ist aufgetreten
144	90	Strom am Eingang ist zu hoch

© ifm electronic gmbh | www.ifm.com

9.2 Zählerfunktionen zur Frequenz- und Periodendauermessung

Inhalt

FASTCOUNT 216
INC_ENCODER 218
PERIOD 221

8110

Je nach Controller werden bis zu 4 schnelle Eingänge unterstützt, die Eingangsfrequenzen bis zu 30 kHz verarbeiten können.

Die Eingänge IN0...IN3 können u.a. für folgende Messungen eingesetzt werden:
- Frequenzmessung,
- Auswertung von inkrementellen Drehgebern.

Bedingt durch die unterschiedlichen Messmethoden können Fehler bei der Frequenzermittlung auftreten.

Zur einfachen Auswertung stehen folgende Bausteine zur Verfügung:

Baustein	Betriebsart	Frequenzbereich [kHz]	Erklärung
FASTCOUNT	Up-Counter	< 30	Schnelle Impulse aufwärts zählen (inkrementieren)
	Down-Counter	< 30	Schnelle Impulse abwärts zählen (dekrementieren)
INC_ENCODER		< 1	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD	Frequenz-Messung	< 30	Messung der Frequenz am angegebenen Kanal Messfehler verringert sich bei hohen Frequenzen.
	Periodendauer-Messung	< 3	Messung der Periodendauer (Zykluszeit) am angegebenen Kanal Hohe Frequenzen verursachen durch eine hohe Interrupt-Last längere Zykluszeiten und eine langsamere Systemreaktion
	Periodendauer- und Ratio-Messung	< 3	Messung der Periodendauer sowie des Puls-Pause-Verhältnis am angegebenen Kanal. Hohe Frequenzen verursachen durch eine sehr hohe Interrupt-Last längere Zykluszeiten und eine langsamere Systemreaktion.

HINWEIS

Wichtig bei Einsatz der schnellen Eingänge als Digitaleingänge:

Werden die schnellen Eingänge als "normale" Digitalänge eingesetzt, muss die erhöhte Empfindlichkeit gegen Störimpulse beachtet werden (z.B. Kontaktprellen bei mechanischen Kontakten). Das Eingangssignal muss bei Bedarf entprellt werden. Der Standard-Digitaleingang kann Signale bis 50 Hz auswerten.

© ifm electronic gmbh | www.ifm.com

9.2.1 FASTCOUNT

8112

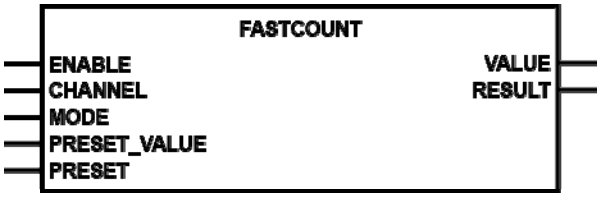
= Fast Count

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8114

FASTCOUNT arbeitet als Zählerbaustein für schnelle Eingangsimpulse.

Dieser FB erfasst Impulse an den schnellen Eingangskanälen (→ Datenblatt).

Parameter der Eingänge

8115

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Eingangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
MODE	BYTE	Betriebsart des Bausteins:
		0_{10} 00_{16} Zähler gestoppt
		21_{10} 15_{16} Aufwärts-Zähler
		22_{10} 16_{16} Abwärts-Zähler
PRESET_VALUE	DWORD	Zähler-Startwert
PRESET	BOOL	TRUE: Startwert PRESET_VALUE wird geladen FALSE: Startwert "0" wird geladen

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8116

Parameter	Datentyp	Beschreibung
VALUE	DWORD	Ausgabewert
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig

© ifm electronic gmbh | www.ifm.com

9.2.2 INC_ENCODER

8134

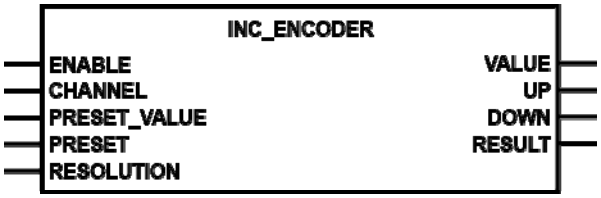
= Incremental Encoder

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8135

INC_ENCODER organisiert Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern.

Immer zwei Frequenzeingänge bilden das Eingangspaar, das mit dem FB ausgewertet wird. In folgender Tabelle finden Sie die zulässigen Grenzfrequenzen und die max. anschließbaren inkrementalen Drehgeber:

Gerät	Grenzfrequenz	max. Anzahl Drehgeber
BasicController: CR040n	1 kHz	2
CabinetController: CR030n	10 kHz	2
ClassicController: CR0020, CR0505	10 kHz	4
ClassicController: CR0032, CR0033	30 kHz	4
ExtendedController: CR0200	10 kHz	8
ExtendedController: CR0232, CR0233	30 kHz	8
Platinensteuerung: CS0015	0,5 kHz	2
SafetyController: CR7020, CR7021, CR7505, CR7506	10 kHz	4
SafetyController: CR7032	30 kHz	4
ExtendedSafetyController: CR7200, CR7201	10 kHz	8
ExtendedSafetyController: CR7132	30 kHz	8
SmartController: CR25nn	10 kHz	2
PDM360smart: CR1071	1 kHz	2

HINWEIS

Je nach weiterer Belastung des Geräts kann die Grenzfrequenz sinken, wenn "viele" Drehgeber ausgewertet werden.

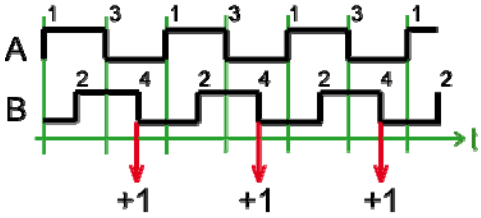
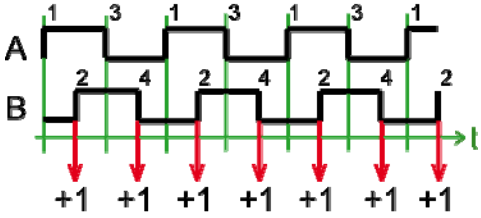
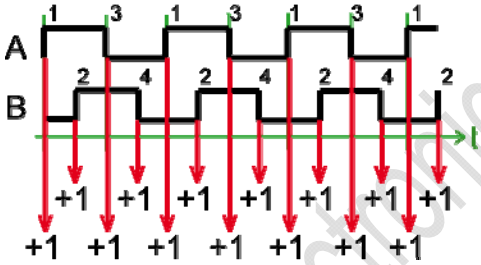
Bei zu hoher Belastung kann die Zykluszeit unzulässig lang werden (→ ***Begrenzungen und Programmierhinweise*** (→ Seite 41)).

© ifm electronic gmbh | www.ifm.com

Über den PRESET_VALUE kann der Zähler auf einen Voreinstellwert gesetzt werden. Der Wert wird übernommen, wenn PRESET auf TRUE gesetzt wird. Anschließend muss PRESET wieder auf FALSE gesetzt werden, damit der Zähler wieder aktiv wird.

Am Ausgang VALUE steht der aktuelle Zählerstand an. Die Ausgänge UP und DOWN zeigen die letzte Zählrichtung des Zählers an. Die Ausgänge sind dann TRUE, wenn der Zähler in die entsprechende Richtung gezählt hat. Wurde der Encoder seit dem letzten Aufruf des Bausteins nicht verändert, sind beide Ausgänge FALSE.

Am Eingang RESOLUTION kann die Auflösung des Drehgebers vervielfacht ausgewertet werden:
1 = normale Auflösung (-536 870 912...536 870 911, identisch mit der Auflösung des Drehgebers),
2 = Auflösung doppelt auswerten (-1 073 741 824...1 073 741 823),
4 = Auflösung 4-fach auswerten (-2 147 483 648...2 147 483 647).
Alle anderen Werte an diesem Eingang bedeuten normale Auflösung.

	RESOLUTION = 1 zählt bei jeder vierten Flanke (= Auflösung des Drehgebers)
	RESOLUTION = 2 zählt bei jeder zweiten Flanke
	RESOLUTION = 4 zählt bei jeder steigenden und fallenden Flanke

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

8137

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Eingangskanal-Paares (0/2): 0 = Kanalpaar 0 = Eingänge 0 + 1 2 = Kanalpaar 1 = Eingänge 2 + 3
PRESET_VALUE	DINT	Zähler-Startwert
PRESET	BOOL	TRUE: Startwert PRESET_VALUE wird geladen FALSE: Zähler aktiv
RESOLUTION	BYTE	Auswertung der Drehgeber-Auflösung: 01 = zählt bei jeder vierten Flanke (= Auflösung des Drehgebers) 02 = zählt bei jeder zweiten Flanke 04 = zählt bei jeder steigenden und fallenden Flanke Alle anderen Werte zählen wie "01".

Parameter der Ausgänge

8138

Parameter	Datentyp	Beschreibung
VALUE	DINT	wenn RESOLUTION = 1: VALUE = –536 870 912...536 870 911 wenn RESOLUTION = 2: VALUE = –1 073 741 824...1 073 741 823 wenn RESOLUTION = 4: VALUE = –2 147 483 648…2 147 483 647
UP	BOOL	Zähler zählt aufwärts
DOWN	BOOL	Zähler zählt abwärts
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
138	8A	Auflösungseinstellung ist ungültig

© ifm electronic gmbh | www.ifm.com

9.2.3 PERIOD

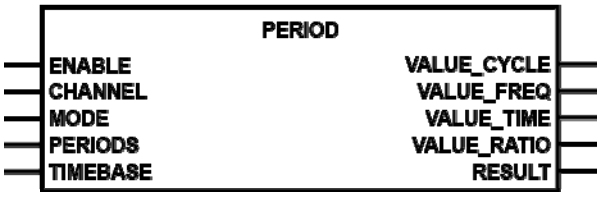
8122

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8123

PERIOD misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal. Maximale Eingangsfrequenz → Datenblatt.

In der Betriebsart Frequenzmessung erfolgt die Messung durch Zählen der positiven Flanken über eine bestimmte Zeit (BASETIME).

In der Betriebsart Periodendauermessung wird der Zeitabstand zwischen zwei positiven Flanken gemessen und der Mittelwert über eine bestimmte Anzahl von Perioden (PERIODS) gemittelt.

Im laufenden Betrieb sollte die Betriebsart MODE nicht geändert werden.

Parameter der Eingänge

8124

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Eingangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
MODE	BYTE	Betriebsart des Bausteins:
		0_{10} 00_{16} keine Messung
		14_{10} $0E_{16}$ Frequenzmessung
		19_{10} 13_{16} Periodendauermessung
		20_{10} 14_{16} Periodendauer- und Ratiomessung
PERIODS	BYTE	Anzahl der zu vergleichenden Perioden zulässige Werte = 1...4
TIMEBASE	TIME	Zeitdauer zum Zählen der Flanken in [ms] zulässige Werte = 1...2 000

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8125

Parameter	Datentyp	Beschreibung
VALUE_CYCLE	DWORD	Periodendauer des Eingangssignals in [µs]
VALUE_FREQ	REAL	Frequenz des Eingangssignals in [Hz]
VALUE_TIME	TIME	Verstrichene Zeit seit der letzten positiven Flanke
VALUE_RATIO	WORD	Puls-/Pause-Verhältnis des Eingangssignals in [‰]
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
137	89	Wert für PERIODS oder TIMEBASE ist ungültig

© ifm electronic gmbh | www.ifm.com

9.3 Ausgangsfunktionen allgemein

10462

Für dieses Gerät können Sie die Funktionsweise von einigen oder von allen Ausgängen einstellen.
Hier zeigen wir Ihnen geeignete Bausteine dazu.

© ifm electronic gmbh | www.ifm.com

9.3.1 OUTPUT

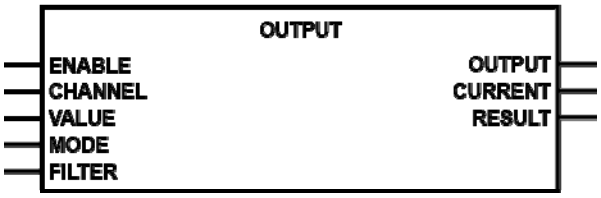
8078

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8079

OUTPUT stellt die Betriebsart von Ausgangskanälen ein.

Im laufenden Betrieb sollte die Betriebsart MODE nicht geändert werden.

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

8080

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Ausgangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
VALUE	BOOL	TRUE: Ausgang aktivieren FALSE: Ausgang deaktivieren
MODE	BYTE	Betriebsart des Bausteins:
		0_{10} 00_{16} Aus
		2_{10} 02_{16} Binärausgang highside
		15_{10} $0F_{16}$ Binärausgang highside mit Diagnose
		16_{10} 10_{16} Binärausgang highside mit Diagnose und Protection
FILTER	BYTE	Filter für die Messung am Eingang
		0_{10} 00_{16} kein Filter
		1_{10} 01_{16} 600 Hz
		2_{10} 02_{16} 233 Hz
		3_{10} 03_{16} 109 Hz
		4_{10} 04_{16} 52 Hz (empfohlen)
		5_{10} 05_{16} 26 Hz
		6_{10} 06_{16} 13 Hz
		7_{10} 07_{16} 6 Hz
		8_{10} 08_{16} 4 Hz

Im laufenden Betrieb sollte die Betriebsart MODE nicht geändert werden.

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8081

Parameter	Datentyp	Beschreibung
OUTPUT	BOOL	TRUE: Ausgang ist aktiviert FALSE: Ausgang ist deaktiviert
CURRENT	WORD	nur für stromgeregelte Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
129	81	Überspannung an VBBx
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
136	88	Filtereinstellung ist ungültig
141	8D	ein Leiterbruch wurde erkannt (bei Binärausgang Highside mit Diagnose)
142	8E	ein Kurzschluss wurde erkannt (bei Binärausgang Highside mit Diagnose)
145	91	Strom am Ausgang ist zu hoch (bei Binärausgang Highside mit Diagnose und Protection)

© ifm electronic gmbh | www.ifm.com

9.4 PWM-Funktionen

Inhalt

Verfügbarkeit von PWM 227
PWM-Signalverarbeitung 228
Stromregelung mit PWM 232

2303

In diesem Kapitel erfahren Sie mehr über die Pulsweitenmodulation im ecomatmobile-Gerät.

9.4.1 Verfügbarkeit von PWM

8472

PWM ist in folgenden Geräten verfügbar:

	Anzahl verfügbare PWM-Ausgänge	davon stromgeregelt (PWMi)	PWM-Frequenz [Hz]
BasicController: CR0401	8	0	20...250
BasicController: CR0403	12	2	20...250
CabinetController: CR0301	4	0	25...250
CabinetController: CR0302, CR0303	8	0	25...250
ClassicController: CR0020	12	8	25...250
ClassicController: CR0505	8	8	25...250
ClassicController: CR0032, CR0033	16	16	25...250
ExtendedController: CR0200	24	16	25...250
ExtendedController: CR0232, CR0233	32	32	25...250
Platinensteuerung: CS0015	8	0	25...250
SafetyController: CR7020, CR7021	12	8	25...250
SafetyController: CR7505, CR0506	8	8	25...250
ExtendedSafetyController: CR7200, CR7201	24	16	25...250
SmartController: CR25nn	4	4	25...250
PDM360smart: CR1071	4	0	25...250

© ifm electronic gmbh | www.ifm.com

9.4.2 PWM-Signalverarbeitung

Inhalt

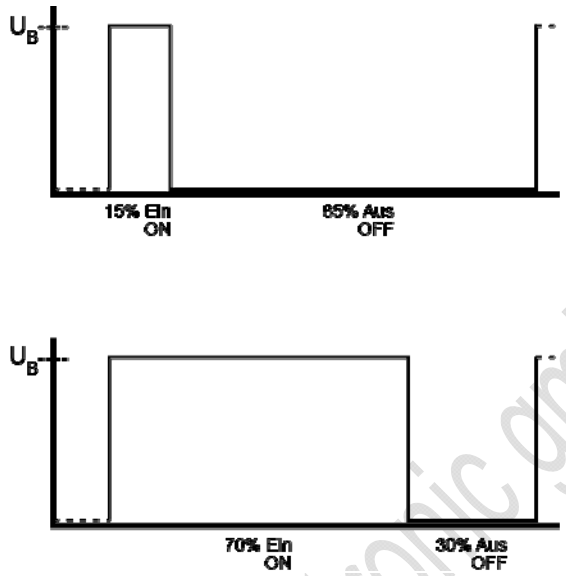
PWM – Einführung 228
PWM-Funktionen und deren Parameter 229

2314

PWM – Einführung

7475
6889

PWM steht als Abkürzung für die **P**uls-**W**eiten-**M**odulation, zuweilen auch "Puls-Pausen-Modulation" (PPM) genannt. Sie wird im Bereich der Steuerungen für den mobilen und robusten Einsatz hauptsächlich zur Ansteuerung von proportionalen Ventilen (PWM-Ventilen) genutzt. Ferner kann durch eine entsprechende Zusatzbeschaltung eines PWM-Ausganges (Zubehör) aus dem pulsweitenmodulierten Ausgangssignal eine analoge Ausgangsspannung erzeugt werden.



Grafik: Prinzip PWM

Bei dem PWM-Ausgangssignal handelt es sich um ein getaktetes Signal zwischen GND und Versorgungsspannung. Innerhalb einer festen Periode (PWM-Frequenz) wird das Puls-/Pausenverhältnis variiert. Durch die angeschlossene Last stellt sich je nach Puls-/Pausenverhältnis der entsprechende Effektivstrom ein.

Die PWM-Funktion der Controller ist eine Hardware-Funktion, die vom Prozessor zur Verfügung gestellt wird. Um die integrierten PWM-Ausgänge des Controllers zu nutzen, müssen diese im Applikations-Programm initialisiert und entsprechend dem gewünschten Ausgangssignal parametriert werden.

HINWEIS

Die PWM-Funktion bleibt solange gesetzt, bis an der Steuerung ein Hardware-Reset durchgeführt wurde → Versorgungsspannung ausschalten und wieder einschalten.

Bei hohen PWM-Frequenzen kann es systembedingt zu Differenzen kommen zwischen eingestelltem und ausgegebenem Ratio-Verhältnis.

© ifm electronic gmbh | www.ifm.com

PWM-Funktionen und deren Parameter

Inhalt

PWM-Frequenz 229
PWM-Kanäle 0...7/11 229
PWM1000 229

1527

PWM-Frequenz

8073

Abhängig vom Ventiltyp wird eine entsprechende PWM-Frequenz benötigt. Die PWM-Frequenz wird bei ***PWM1000*** (→ Seite 229) direkt als Zahlenwert in [Hz] übergeben.

Alle PWM-Kanäle verhalten sich gleich. Jeder PWM-Kanal kann unabhängig auf eine eigene Frequenz eingestellt werden. Die PWM-Frequenz liegt im Bereich 20...250 Hz.

PWM-Kanäle 0...7/11

8074

Für folgende Geräte verfügbar:

BasicController: CR0401	PWM-Kanäle 0...7
BasicController: CR0403	PWM-Kanäle 0...11

Alle PWM-Kanäle bieten die größte Flexibilität bei der Parametrierung. Für jeden Kanal kann jeweils eine eigene PWM-Frequenz und das Puls-Pause-Verhältnis eingestellt werden.

Im laufenden Betrieb kann sowohl das Puls-Pause-Verhältnis als auch die PWM-Frequenz geändert werden.

HINWEIS

PWM-Ausgänge dürfen NICHT parallel betrieben werden, um z.B. den max. Ausgangsstrom zu erhöhen. Die Ausgänge arbeiten nicht synchron.

Andernfalls kann die komplette Last über nur einen Ausgang gehen. Die Strommessung funktioniert dann nicht mehr.

© ifm electronic gmbh | www.ifm.com

PWM1000

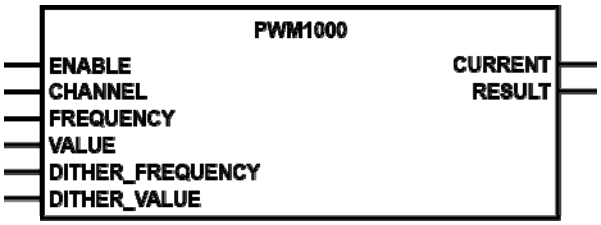
8060

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n

Symbol in CoDeSys:



Beschreibung

8062

PWM1000 dient der Initialisierung und Parametrierung der PWM-Ausgänge.

Der FB ermöglicht eine einfache Anwendung der PWM-Funktion im Gerät. Für jeden Kanal kann jeweils eine eigene PWM-Frequenz und das Puls-Pause-Verhältnis eingestellt werden.

Die PWM-Frequenz FREQUENCY kann direkt in [Hz] und das Puls-Pause-Verhältnis VALUE in 1 ‰-Schritten angegeben werden.

Parameter der Eingänge

8063

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Ausgangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
FREQUENCY	WORD	PWM-Frequenz in [Hz] zulässige Werte = 0014...00FA$_{16}$ = 20...250$_{10}$
VALUE	WORD	PWM-Wert (Puls-Pause-Verhältnis) in [‰] zulässige Werte = 0000...03E8$_{16}$ = 0...1 000$_{10}$
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY sollte geradzahlig sein
DITHER_VALUE	WORD	Amplitude des Dither-Wertes in [‰] zulässige Werte = 0000...03E8$_{16}$ = 0...1 000$_{10}$

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8523

Parameter	Datentyp	Beschreibung
CURRENT	WORD	nur für stromgeregelte Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
129	81	Überspannung an VBBx
130	82	Kanaleinstellung ist ungültig
131	83	Wert für VALUE ist ungültig
133	85	Wert für FREQUENCY ist ungültig
134	86	Dither-Einstellung ist ungültig

© ifm electronic gmbh | www.ifm.com

9.4.3 Stromregelung mit PWM

Inhalt

Überlastschutz 232
Strommessung bei PWM-Kanälen 232
CURRENT_CONTROL 233

1550

Dieses Gerät kann den tatsächlich fließenden Strom an bestimmten Ausgängen messen und das Signal zur Weiterverarbeitung nutzen. **ifm electronic** stellt dem Anwender einige Bausteine zu diesem Zweck bereit.

Überlastschutz

8524

Grundsätzlich sind die stromgeregelten Ausgänge gegen Kurzschluss und Überlast geschützt, da die Regelung versucht, den Strom zu reduzieren.

ACHTUNG

Die Messwiderstände (Shunts) könnten im Überlastfall thermisch überlastet werden, wenn...
- der Ausgang als Binärausgang betrieben wird und
- die Ströme z.B. zwischen 8 A und 20 A durch Leitungslängen und -querschnitte begrenzt sind.

► Daher im Applikations-Programm bei Verwendung des FB OUTPUT die Betriebsart [Binärausgang Highside mit Diagnose und Protection] an diesen Ausgängen wählen.

> Bei Strömen > 2,25 A wird der betroffene Ausgang automatisch abgeschaltet.

> Ist der Ausgang nicht mehr überlastet, wird der Ausgang selbsttätig wieder eingeschaltet.

Ein Überlastschutz ist im reinen PWM-Modus (ohne Stromregelung) NICHT aktiv!

Strommessung bei PWM-Kanälen

8076

Über die im Controller integrierten Strommesskanäle kann eine Strommessung des Spulenstroms durchgeführt werden. Dadurch kann zum Beispiel der Strom bei einer Spulenerwärmung nachgeregelt werden. Damit bleiben die Hydraulikverhältnisse im System gleich.

Bei PWM-Frequenzen unter 100 Hz und zusätzlichem Dither kann die Stromregelung die angegebene Genauigkeit (→ Datenblatt) nicht mehr erreichen.

© ifm electronic gmbh | www.ifm.com

CURRENT_CONTROL

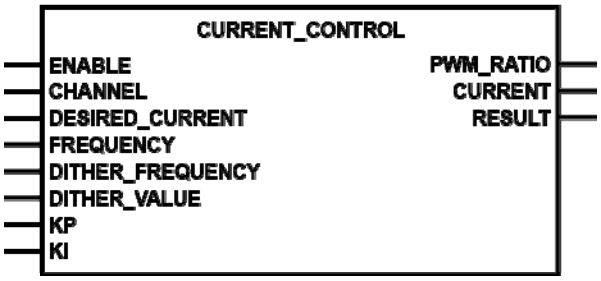
8082

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR0403

Symbol in CoDeSys:



Beschreibung

8086

CURRENT_CONTROL arbeitet als Stromregler für die PWMi-Ausgänge.

Der Regler regelt in Abhängigkeit der Periodendauer des PWM-Signals. Die beiden Anstellparameter KI und KP repräsentieren den **I**ntegral- und den **P**roportional-Anteil des Reglers.

► Zur Ermittlung der besten Einstellung des Reglers bietet sich als Startwert an, KI=50 und KP=50 zu setzen. Je nach gewünschtem Reglerverhalten können die Werte schrittweise vergrößert (Regler wird härter / schneller) oder verkleinert (Regler wird schwächer / langsamer) werden.

> Bei Sollwert DESIRED_CURRENT=0 wird der Ausgang **sofort** auf 0 mA geschaltet, wobei **nicht** entsprechend der eingestellten Parameter auf 0 mA heruntergeregelt wird.

Der Regler verfügt über einen schnellen Ausgleichsmechanismus bei Spannungseinbrüchen der Versorgungsspannung. In Abhängigkeit der Größe des Spannungseinbruchs wird zusätzlich zum Regelverhalten des Reglers die Ratio des PWMs dementsprechend so vergrößert, dass der Regler so schnell wie möglich den Sollwert erreicht.

Je nach eingesetzter Steuerungs-Hardware ist ein unterschiedliches Teach-Verhalten zu beachten.

HINWEIS

► Bei der Definition des Parameters DITHER_VALUE darauf achten, dass das resultierende PWM-Ratio im Arbeitsbereich der Regelung zwischen 0...1000 ‰ bleibt:
- PWM-Ratio + DITHER_VALUE < 1000 ‰ und
- PWM-Ratio - DITHER_VALUE > 0 ‰.

Außerhalb dieses zulässigen Bereichs kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden.

> Kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden, weil das PWM-Ratioverhältnis bereits bei 1000 ‰ ist, wird das im Ausgang RESULT angezeigt.

Bei PWM-Frequenzen unter 100 Hz und zusätzlichem Dither kann die Stromregelung die angegebene Genauigkeit (→ Datenblatt) nicht mehr erreichen.

ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

8087

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
CHANNEL	BYTE	Nummer des Ausgangskanals (0...x Wert abhängig vom Gerät, → Datenblatt)
DESIRED_CURRENT	WORD	Stromsollwert des Ausgangs in [mA] (0...x Wert abhängig vom Gerät, → Datenblatt)
FREQUENCY	WORD	zulässige PWM-Frequenz in [Hz] für die am Ausgang angeschlossene Last zulässige Werte = 0014...00FA$_{16}$ = 20...250$_{10}$
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY sollte geradzahlig sein
DITHER_VALUE	WORD	Amplitude des Dither-Wertes in [‰] zulässige Werte = 0000...03E8$_{16}$ = 0...1 000$_{10}$
KP	BYTE	Proportional-Anteil des Ausgangsignals
KI	BYTE	Integral-Anteil des Ausgangsignals

Parameter der Ausgänge

8088

Parameter	Datentyp	Beschreibung
PWM_RATIO	WORD	Zu Kontrollzwecken: Anzeige PWM-Tastverhältnis 0...1000 ‰
CURRENT	WORD	nur für stromgeregelte Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
003	03	Funktionsblock ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
129	81	Überspannung an VBBx
130	82	Kanaleinstellung ist ungültig
131	83	Wert für DESIRED_CURRENT ist ungültig
134	86	Dither-Einstellung ist ungültig

© ifm electronic gmbh | www.ifm.com

10 Daten verwalten

Inhalt

Systemzeit lesen / schreiben 235
Daten im Speicher sichern, lesen und wandeln 237
Datenzugriff und Datenprüfung 244
Dateien verwalten 248

8606

Hier zeigen wir Ihnen Funktionen, mit denen Sie Daten im Gerät lesen und verarbeiten können.

10.1 Systemzeit lesen / schreiben

Inhalt

TIMER_READ_US 236

1601

Mit folgenden Bausteinen der **ifm electronic gmbh** können Sie die kontinuierlich laufende Systemzeit des Controllers lesen und im Applikations-Programm auswerten oder bei Bedarf ändern.

© ifm electronic gmbh | www.ifm.com

10.1.1 TIMER_READ_US

8219

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

660

TIMER_READ_US liest die aktuelle Systemzeit in [µs] aus.

Mit Anlegen der Versorgungsspannung bildet das Gerät einen Zeittakt, der in einem Register aufwärts gezählt wird. Dieses Register kann mittels des FB-Aufrufes ausgelesen werden und z.B. zur Zeitmessung genutzt werden.

Info

Der System-Timer läuft maximal bis zum Zählerwert 4294967295 (µs) und startet anschließend wieder bei 0.

4 294 967 295 µs = 4 295 s = 71,6 min = 1,2 h

Parameter der Ausgänge

8220

Parameter	Datentyp	Beschreibung
TIME_US	DWORD	Aktuelle Systemzeit (Auflösung [µs])
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh | www.ifm.com

10.2 Daten im Speicher sichern, lesen und wandeln

Inhalt

Automatische Datensicherung 237
Manuelle Datensicherung..... 238

1595

10.2.1 Automatische Datensicherung

9237

Die ecomatmobile-Geräte bieten die Möglichkeit, Daten (BOOL, BYTE, WORD, DWORD) remanent (= spannungsausfallsicher) im Speicher zu sichern. Voraussetzung ist, dass die Daten als RETAIN-Variablen angelegt wurden (→ CoDeSys).

Man unterscheidet zwischen Variablen, die als RETAIN deklariert wurden, und Variablen im Merkerbereich.

Der Vorteil des automatischen Speicherns ist, dass auch bei einem plötzlichen Spannungsabfall oder einer Unterbrechung der Versorgungsspannung die aktuellen Werte der Daten erhalten bleiben (z.B. Zählerstände).

Unterhalb einer Versorgungsspannung (VBBs) von ca. 8,5V werden die Retain-Daten NICHT gesichert. Es kann dann nicht mehr sichergestellt werden, dass alle Daten bei einem Verlust der Versorgungsspannung gesichert werden.

Als VAR_RETAIN deklarierte Variablen werden nach folgenden Ereignissen aus dem Retain-Speicher geladen:
- Power OFF ⇨ ON
- Reset warm

Nach folgenden Ereignissen werden die als VAR_RETAIN deklarierten Variablen neu initialisiert:
- Reset kalt
- Reset Ursprung
- Applikationsprogramm laden
- Laufzeitsystem laden

Variablen im Merkerbereich (ab %MB0) werden niemals neu initialisiert.

© ifm electronic gmbh | www.ifm.com

10.2.2 Manuelle Datensicherung

Inhalt

FLASH_INFO 239
FLASH_READ 239
MEMCPY 241

1597

Neben der Möglichkeit, die Daten automatisch zu sichern, können über FB-Aufrufe Anwenderdaten manuell in integrierte Speicher gesichert und von dort wieder gelesen werden.

Je nach Gerät stehen folgende Speicher zur Verfügung:

Speicher / Gerät	Eigenschaften
EEPROM-Speicher Für folgende Geräte verfügbar: - CabinetController: CR0301, CR0302 - Platinensteuerung: CS0015 - SmartController: CR25nn	Langsames Schreiben und Lesen. Begrenzte Schreib-/Lesehäufigkeit. Beliebige Speicherbereiche wählbar. Daten sichern mit E2WRITE. Daten lesen mit E2READ
FRAM-Speicher [1]) Für folgende Geräte verfügbar: - CabinetController: CR0303 - ClassicController: CR0020, CR0032, CR0033, CR0505 - ExtendedController: CR0200, CR0232, CR0233 - SafetyController: CR7nnn - PDM360smart: CR1070, CR1071	Schnelles Schreiben und Lesen. Unbegrenzte Schreib-/Lesehäufigkeit. Beliebige Speicherbereiche wählbar. Daten sichern mit FRAMWRITE. Daten lesen mit FRAMREAD.
Flash-Speicher Für alle Geräte	Schnelles Schreiben und Lesen. Begrenzte Schreib-/Lesehäufigkeit. Nur zum Speichern großer Datenmengen sinnvoll einsetzbar. Vor dem erneuten Schreiben muss Speicherinhalt gelöscht werden. Daten sichern mit FLASHWRITE. Daten lesen mit FLASHREAD.

[1]) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

Der Programmierer kann sich anhand der Speicheraufteilung (→ Kapitel ***Verfügbarer Speicher*** (→ Seite 43) darüber informieren, welcher Speicherbereich frei zur Verfügung steht.

© ifm electronic gmbh | www.ifm.com

FLASH_INFO

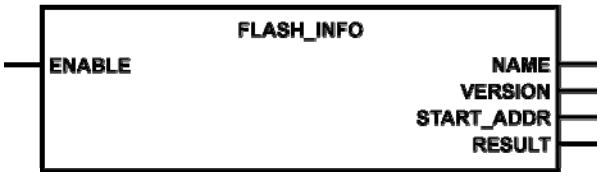
11580

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:



Beschreibung

11588

FLASH_INFO liest die Informationen aus dem User-Flash-Speicher:
- Name des Speicherbereichs (vom User vorgegeben),
- Software-Version,
- Startadresse (für einfaches Lesen mit IEC-Struktur).

Details und Beispiele → Kapitel ***Arbeiten mit dem User-Flash-Speicher*** (→ Seite 279)

Parameter der Eingänge

11589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv

Parameter der Ausgänge

11590

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name des Speicherbereichs (vom User vorgegeben)
VERSION	STRING(24)	Software-Version
START_ADDR	DWORD	Startadresse der Daten
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
157	9D	Software-Header ist ungültig (CRC-Fehler)

© ifm electronic gmbh | www.ifm.com

FLASH_READ

8147

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

11579

FLASH_READ ermöglicht das Lesen unterschiedlicher Datentypen direkt aus dem Flash-Speicher.

Der FB liest den Inhalt ab der Adresse von SRC aus dem Flash-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.

► Für die Zieladresse DST gilt: Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Details und Beispiele → Kapitel ***Arbeiten mit dem User-Flash-Speicher*** (→ Seite 279)

Parameter der Eingänge

8148

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
DST	DWORD	Ziel-Adresse ► Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	Quell-Adresse
LEN	WORD	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 0001...$FFFF_{16}$ = 1...65 535_{10}

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8152

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152	98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes

© ifm electronic gmbh | www.ifm.com

MEMCPY

8160

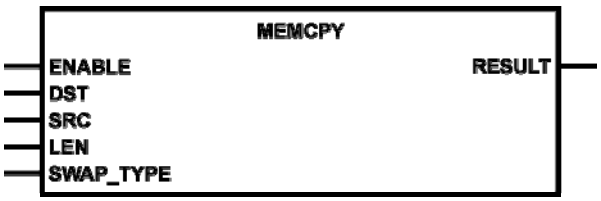
= Memory Copy

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:



Beschreibung

412

MEMCPY ermöglicht das Schreiben und Lesen unterschiedlicher Datentypen direkt in den Speicher.

Der FB schreibt den Inhalt ab der Adresse von SRC an die Adresse DST.

► Für die Zieladresse DST gilt: Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

> Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben wurden. Dadurch ist es auch möglich, genau ein Byte einer Wortdatei zu übertragen.

Parameter der Eingänge

8162

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
DST	DWORD	Ziel-Adresse ► Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	Quell-Adresse
LEN	WORD	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 0001...FFFF$_{16}$ = 1...65 535$_{10}$
SWAP_TYPE	BYTE	Byte-Reihenfolge tauschen: 0 = kein Tausch z.B.: 1A 2B 3C 4D ⇨ 1A 2B 3C 4D 1 = 2 Bytes tauschen (WORD, INT, ...) z.B.: 1A 2B 3C 4D ⇨ 2B 1A 4D 3C LEN muss ein Mehrfaches von 2 sein! 2 = 4 Bytes tauschen (DWORD, DINT, REAL, TIME, ...) z.B.: 1A 2B 3C 4D ⇨ 4D 3C 2B 1A LEN muss ein Mehrfaches von 4 sein!

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

8163

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152	98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes
156	9C	unzulässiger Speicherbereich: • ungültiger Wert für SWAP_TYPE • LEN passt nicht zu SWAP_TYPE

© ifm electronic gmbh | www.ifm.com

10.3 Datenzugriff und Datenprüfung

Inhalt

GET_IDENTITY 245
SET_IDENTITY 246
SET_PASSWORD 247

1598

Die Bausteine in diesem Kapitel steuern den Datenzugriff und ermöglichen ein Prüfen der Daten.

© ifm electronic gmbh | www.ifm.com

10.3.1 GET_IDENTITY

8166

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

2344

GET_IDENTITY liest die im Gerät gespeicherten Geräte- und applikations-spezifischen Kennungen.

Der Name der Applikation kann mit ***SET_IDENTITY*** (→ Seite 246) verändert werden.

Parameter der Eingänge

8167

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv

Parameter der Ausgänge

8168

Parameter	Datentyp	Beschreibung
APP_IDENT	STRING(80)	Name der Applikation als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704"
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
155	9B	Wert konnte nicht gelesen werden

© ifm electronic gmbh | www.ifm.com

10.3.2 SET_IDENTITY

8174

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

8535

SET_IDENTITY setzt eine applikationsspezifische Programmkennung.

Mit dem FB kann durch das Applikations-Programm eine Programmkennung erzeugt werden. Diese Kennung kann zur Identifizierung des geladenen Programms über das Maintenance-Tool als Software-Version ausgelesen werden.

Parameter der Eingänge

8175

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
APP_IDENT	STRING(80)	Name der Applikation als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704" Rücksetzen mit APP_IDENT = ""

Parameter der Ausgänge

8176

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

ifm electronic gmbh | www.ifm.com

10.3.3 SET_PASSWORD

8178

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

8179

SET_PASSWORD setzt Benutzerkennung für Programm- und Speicher-Upload mit dem Maintenance-Tool.

Ist die Benutzerkennung aktiv, kann durch das Maintenance-Tool das Applikations-Programm oder der Datenspeicher nur ausgelesen werden, wenn das richtige Password eingegeben wurde.

Wird an den Eingang PASSWORD ein Leer-String (Default-Zustand) übergeben, ist ein Upload der Applikations-Software oder des Datenspeichers jederzeit möglich.

Beim Laden eines neuen Applikations-Programms wird die Kennung wieder zurückgesetzt.

Parameter der Eingänge

8180

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
PASSWORD	STRING(16)	Benutzerkennung Wenn PASSWORD = "", dann ist Zugriff ohne Passworteingabe möglich.

Parameter der Ausgänge

8181

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh | www.ifm.com

10.4 Dateien verwalten

Inhalt

Dateifunktionen 248

8611

Hier zeigen wir Ihnen Funktionen, mit denen Sie Dateien und ihre Inhalte verwalten können.

10.4.1 Dateifunktionen

Inhalt

GET_APP_INFO 249
GET_HW_INFO 249
GET_SW_INFO 251

11525

© ifm electronic gmbh | www.ifm.com

GET_APP_INFO

11581

= Get Application Information

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

11593

GET_APP_INFO liefert Informationen über die im Gerät gespeicherte Applikations-Software:
- Name der Applikation,
- Version der Applikation,
- eindeutige CoDeSys-Build-Nummer,
- CoDeSys-Build-Datum.

Parameter der Eingänge

11594

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv

Parameter der Ausgänge

11595

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der Applikation
VERSION	STRING(24)	Version
BUILD_NUM	STRING(24)	eindeutige CoDeSys-Build-Nummer
BUILD_DATE	STRING(24)	CoDeSys-Build-Datum
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh | www.ifm.com

GET_HW_INFO

11582

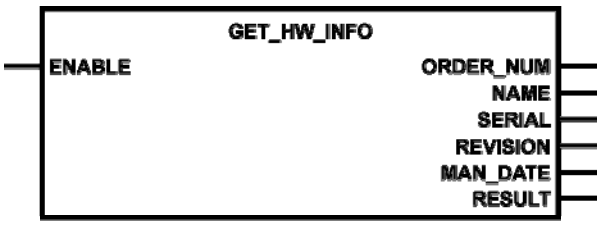
= Get Hardware Information

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:



Beschreibung

1599

GET_HW_INFO liefert Informationen über die Hardware des Geräts:
- ifm-Artikelnummer (z.B. CR0403),
- Artikelbezeichnung,
- eindeutige Seriennummer,
- Hardware-Revision,
- Produktionsdatum.

Parameter der Eingänge

11600

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv

© ifm electronic gmbh | www.ifm.com

Parameter der Ausgänge

11601

Parameter	Datentyp	Beschreibung
ORDER_NUM	STRING(24)	ifm-Artikelnummer (z.B.: CR0403)
NAME	STRING(24)	Artikelbezeichnung (z.B.: "BasicController 12/12")
SERIAL	STRING(24)	Seriennummer des Geräts (z.B.: "000045784")
REVISION	STRING(24)	Hardware-Revisionsstand des Geräts (z.B.: "AB")
MAN_DATE	STRING(24)	Herstellungsdatum des Geräts (z.B.: "20111007123800")
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh | www.ifm.com

GET_SW_INFO

11583

= Get Software Information

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

11596

GET_SW_INFO liefert Informationen über die System-Software des Geräts:
- Software-Name,
- Software-Version,
- Build-Nummer,
- Build-Datum.

Parameter der Eingänge

11597

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv

Parameter der Ausgänge

11598

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der Software (z.B.: "BasicSystem")
VERSION	STRING(24)	Version der Software (z.B.: "V02.00.03")
BUILD_NUM	STRING(24)	Build-Nummer (z.B.: "45")
BUILD_DATE	STRING(24)	Build-Datum (z.B.: "20111006123800")
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh | www.ifm.com

11 SPS-Zyklus optimieren

Inhalt

Zykluszeit steuern 253
8609

Hier zeigen wir Ihnen Funktionen zum Optimieren des SPS-Zyklus.

11.1 Zykluszeit steuern

Inhalt

PLCPRG_DELAY 254
3142

© ifm electronic gmbh | www.ifm.com

11.1.1 PLCPRG_DELAY

11655

Die Zeit zwischen zwei aufeinander folgenden PLC_PRG-Zyklen ist einstellbar. Dazu gibt es in der Steuerungskonfiguration den Parameter PLCPRG_DELAY.

- BasicController: CR040n
  Somit kann die Systemauslastung zwischen Visualisierung und PLC_PRG angepasst werden. Der Applikations-Programmierer legt hier fest, ob er die Priorität auf der Visualisierung oder auf der PLC_PRG haben will.
- BasicDisplay: CR0451
  Somit kann eine Wartezeit zwischen den einzelnen Zyklen eingestellt werden.

Einstellung in CR040n

11656

In CoDeSys > Reiter [Ressourcen] > [Steuerungskonfiguration]
> [CR040n Configuration Vnn] > [Device Configuration(FIX)]
> Reiter [Modulparameter]

Eine Tabelle erscheint mit folgenden Elementen:

Spalte	Wert	Bedeutung
Index	001	lfd. Nr. in dieser Liste
Name	PLCPRG_DELAY	Name des Parameters
Wert	0	aktueller Wert dieses Parameters in [ms]
Default	0	werksseitig voreingestellter Wert dieses Parameters in [ms]
Min.	0	Mindestwert dieses Parameters in [ms]
Max.	90	Maximalwert dieses Parameters in [ms]

► Bei Bedarf den Wert in der Spalte [Wert] überschreiben.
Die anderen Einträge in dieser Zeile NICHT ändern!

► Das CoDeSys-Projekt speichern
das Projekt neu übersetzen,
ein Bootprojekt erzeugen,
das Projekt in das Gerät übertragen.

© ifm electronic gmbh | www.ifm.com

Einstellung in CR0451

11657

In CoDeSys > Reiter [Ressourcen] > [Steuerungskonfiguration]
> [CR0451 Configuration Vnn] > [Device Configuration(FIX)]
> Reiter [Modulparameter]

Eine Tabelle erscheint mit folgenden Elementen:

Spalte	Wert	Bedeutung
Index	001	lfd. Nr. in dieser Liste
Name	PLCPRG_DELAY	Name des Parameters
Wert	10	aktueller Wert dieses Parameters in [ms] Je größer [Wert], desto mehr Priorität das das PLC-Programm gegenüber der Visualisierung.
Default	10	werksseitig voreingestellter Wert dieses Parameters in [ms]
Min.	5	Mindestwert dieses Parameters in [ms]
Max.	90	Maximalwert dieses Parameters in [ms]

► Bei Bedarf den Wert in der Spalte [Wert] überschreiben.
Die anderen Einträge in dieser Zeile NICHT ändern!

► Das CoDeSys-Projekt speichern
das Projekt neu übersetzen,
ein Bootprojekt erzeugen,
das Projekt in das Gerät übertragen.

© ifm electronic gmbh | www.ifm.com

12 LED, Buzzer, Visualisierung

Inhalt

LED ansteuern 256
8615

Hier zeigen wir Ihnen folgende Funktionen:

- Ansteuern von LED
- Ansteuern des Buzzer
- Verwalten der Visualisierung

12.1 LED ansteuern

Inhalt

SET_LED..... 257
8230

Hier zeigen wir Ihnen Funktionen zum Ansteuern der LED in diesem Gerät.

© ifm electronic gmbh | www.ifm.com

12.1.1 SET_LED

Inhalt

Beschreibung 257
Parameter der Eingänge 258
Parameter der Ausgänge 258

8052

Baustein-Typ = Funktionsblock (FB)

Enthalten in Bibliothek: `ifm_CRnnnn_Vxxyyzz.LIB`

Für folgende Geräte verfügbar:
- BasicController: CR040n
- BasicDisplay: CR0451

Symbol in CoDeSys:

Beschreibung

8054

Mit SET_LED können im Applikations-Programm Frequenz und Farbe der Status-LED geändert werden.

Wird der Blinkmodus im Applikations-Programm geändert, gilt die Voreinstellung-Tabelle nicht mehr (→ Kapitel ***Status-LED*** (→ Seite 38)).

© ifm electronic gmbh | www.ifm.com

Parameter der Eingänge

8223

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein wird ausgeführt FALSE: Baustein wird nicht ausgeführt > Baustein-Ein- und Ausgänge sind nicht aktiv
COLOR_1	BYTE	LED-Farbe für eingeschaltet Farbkonstante aus GLOBALE_VARIABLEN (CONSTANT) "System LED Colour"; zulässig: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
COLOR_2	BYTE	LED-Farbe für ausgeschaltet Farbkonstante aus GLOBALE_VARIABLEN (CONSTANT) "System LED Colour"; zulässig: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
FREQUENCY	BYTE	LED-Blinkfrequenz Frequenz-Konstante aus GLOBALE VARIABLEN (CONSTANT) "System LED Frequency"; zulässig: 00 = LED_0HZ = dauernd EIN 01 = LED_05HZ = blinkt mit 0,5 Hz 02 = LED_1Hz = blinkt mit 1 Hz 04 = LED_2HZ = blinkt mit 2 Hz 10 = LED_5HZ = blinkt mit 5 Hz

Parameter der Ausgänge

8227

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsblocks (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
000	00	Funktionsblock ist nicht aktiv
001	01	Funktionsblock-Ausführung ohne Fehler beendet
002	02	Funktionsblock ist aktiv (Aktion noch nicht beendet)
133	85	Wert für FREQUENCY ist ungültig
151	97	Wert für Farbe ist ungültig

© ifm electronic gmbh www.ifm.com

13 Anhang

Inhalt

Adressbelegung und E/A-Betriebsarten ... 259
Systemmerker ... 264
CANopen-Tabellen ... 265
Arbeiten mit dem User-Flash-Speicher ... 279
Übersicht der verwendeten Dateien und Bibliotheken ... 294

1664

Hier stellen wir Ihnen – ergänzend zu den Angaben in den Datenblättern – zusammenfassende Tabellen zur Verfügung.

13.1 Adressbelegung und E/A-Betriebsarten

Inhalt

Adressbelegung Ein-/Ausgänge ... 260
Mögliche Betriebsarten Ein-/Ausgänge (CR0401) ... 262
Mögliche Betriebsarten Ein-/Ausgänge (CR0403) ... 263

1656

→ auch Datenblatt

© ifm electronic gmbh | www.ifm.com

13.1.1 Adressbelegung Ein-/Ausgänge

Inhalt

Adressbelegung der Eingänge 260
Adressbelegung der Ausgänge (CR0401) 261
Adressbelegung der Ausgänge (CR0403) 261

2371

Adressbelegung der Eingänge

8538

Abkürzungen → Kapitel ***Hinweise zur Anschlussbelegung*** (→ Seite 33)

Betriebsarten der Ein- und Ausgänge → Kapitel ***Betriebsarten*** (→ Seite 262, → Seite 263)

Nur in der Betriebsart "Analog-Eingang" können die anderen Werte "IN_VOLTAGE10" usw. eingestellt werden.

IEC-Adresse	Name E/A-Variable	Konfiguration mit FB	Default-Modus	mögliche Betriebsarten
%IB00	IN00	INPUT, PERIOD, FASTCOUNT, INC_ENCODER	BL	BL / BH / A (U/I) / FRQ / CYL
%IB01	IN01	INPUT, PERIOD, FASTCOUNT, INC_ENCODER	BL	BL / BH / A (U/I) / FRQ / CYL
%IB02	IN02	INPUT, PERIOD, FASTCOUNT, INC_ENCODER	BL	BL / BH / A (U/I) / FRQ / CYL
%IB03	IN03	INPUT, PERIOD, FASTCOUNT, INC_ENCODER	BL	BL / BH / A (U/I) / FRQ / CYL
%IB04	IN04	INPUT	BL	BL / Widerstandsmessung
%IB05	IN05	INPUT	BL	BL / Widerstandsmessung
%IB06	IN06	INPUT	BL	BL / Widerstandsmessung
%IB07	IN07	INPUT	BL	BL / Widerstandsmessung
%IB08	IN08	INPUT	BL	BL
%IB09	IN09	INPUT	BL	BL
%IB10	IN10	INPUT	BL	BL
%IB11	IN11	INPUT	BL	BL

© ifm electronic gmbh | www.ifm.com

Adressbelegung der Ausgänge (CR0401)

8358

Abkürzungen → Kapitel ***Hinweise zur Anschlussbelegung*** (→ Seite 33)

Betriebsarten der Ein- und Ausgänge → Kapitel ***Betriebsarten*** (→ Seite 262, → Seite 263)

IEC-Adresse	Name E/A-Variable	Konfiguration mit FB	Default-Modus	mögliche Betriebsarten
%QB0	OUT00	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB1	OUT01	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB2	OUT02	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB3	OUT03	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB4	OUT04	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB5	OUT05	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB6	OUT06	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB7	OUT07	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM

Adressbelegung der Ausgänge (CR0403)

8880

Abkürzungen → Kapitel ***Hinweise zur Anschlussbelegung*** (→ Seite 33)

Betriebsarten der Ein- und Ausgänge → Kapitel ***Betriebsarten*** (→ Seite 262, → Seite 263)

IEC-Adresse	Name E/A-Variable	Konfiguration mit FB	Default-Modus	mögliche Betriebsarten
%QB0	OUT00	OUTPUT, PWM1000, CURRENT_CONTROL	H-digital	Aus / H-digital / PWM / PWMi
%QB1	OUT01	OUTPUT, PWM1000, CURRENT_CONTROL	H-digital	Aus / H-digital / PWM / PWMi
%QB2	OUT02	OUTPUT, PWM1000, CURRENT_CONTROL	H-digital	Aus / H-digital / PWM
%QB3	OUT03	OUTPUT, PWM1000, CURRENT_CONTROL	H-digital	Aus / H-digital / PWM
%QB4	OUT04	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB5	OUT05	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB6	OUT06	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB7	OUT07	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB8	OUT08	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB9	OUT09	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB10	OUT10	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM
%QB11	OUT11	OUTPUT, PWM1000	H-digital	Aus / H-digital / PWM

© ifm electronic gmbh | www.ifm.com

13.1.2 Mögliche Betriebsarten Ein-/Ausgänge (CR0401)

8460

Mögliche Konfigurations-Kombinationen (wo zulässig) entstehen durch Addition der Konfigurations-Werte.

Eingänge	Betriebsart für MODE [1])	Konfig.-Wert	Ausgänge	Betriebsart für MODE [2])	Konfig.-Wert
IN00...IN03	deaktiviert	0	OUT00...OUT03	Aus	0
	Spannungseingang 0...10 000 mV	3		Binärausgang Highside	2
	Spannungseingang ratiometrisch 0...1 000 ‰	6			
	Stromeingang 0...20 000 mA	7			
	Spannungseingang 0...32 000 mV	9			
	Binäreingang Lowside	10 (Default)			
	Binäreingang Lowside mit Diagnose (Namur)	11			
	Binäreingang Highside	12			
	Frequenzmessung	14			
				Binärausgang Highside mit Diagnose	15
	Periodendauermessung	19			
	Periodendauer- und Ratiomessung	20			
	Aufwärts-Zähler	21			
	Abwärts-Zähler	22			
IN04...IN07	deaktiviert	0	OUT04...OUT07	Aus	0
				Binärausgang Highside	2
	Binäreingang Lowside	10 (Default)			
	Binäreingang Lowside mit Diagnose (Namur)	11			
				Binärausgang Highside mit Diagnose	15
	Widerstandseingang 0…3 600 Ω	18			
IN08...IN11	deaktiviert	0			
	Binäreingang Lowside	10 (Default)			
	Binäreingang Lowside mit Diagnose (Namur)	11			

[1]) Eingang MODE der FBs INPUT, FASTCOUNT, PERIOD

[2]) Eingang MODE des FBs OUTPUT

© ifm electronic gmbh | www.ifm.com

13.1.3 Mögliche Betriebsarten Ein-/Ausgänge (CR0403)

8539

Mögliche Konfigurations-Kombinationen (wo zulässig) entstehen durch Addition der Konfigurations-Werte.

Eingänge	Betriebsart für MODE [1])	Konfig.-Wert	Ausgänge	Betriebsart für MODE [2])	Konfig.-Wert
IN00...IN03	deaktiviert	0	OUT00...OUT03	Aus	0
	Spannungseingang 0...10 000 mV	3		Binärausgang Highside	2
	Spannungseingang ratiometrisch 0...1 000 ‰	6			
	Stromeingang 0...20 000 mA	7			
	Spannungseingang 0...32 000 mV	9			
	Binäreingang Lowside	10 (Default)		Binärausgang Highside mit Diagnose und Protection	10
	Binäreingang Lowside mit Diagnose (Namur)	11			
	Binäreingang Highside	12			
	Frequenzmessung	14			
				Binärausgang Highside mit Diagnose	15
	Periodendauermessung	19			
	Periodendauer- und Ratiomessung	20			
	Aufwärts-Zähler	21			
	Abwärts-Zähler	22			
IN04...IN07	deaktiviert	0	OUT04...OUT07	Aus	0
				Binärausgang Highside	2
	Binäreingang Lowside	10 (Default)			
	Binäreingang Lowside mit Diagnose (Namur)	11			
				Binärausgang Highside mit Diagnose	15
	Widerstandseingang 0...3 600 Ω	18			
IN08...IN11	deaktiviert	0	OUT08...OUT11	Aus	0
				Binärausgang Highside	2
	Binäreingang Lowside	10 (Default)			
	Binäreingang Lowside mit Diagnose (Namur)	11			
				Binärausgang Highside mit Diagnose	15

[1]) Eingang MODE der FBs INPUT, FASTCOUNT, PERIOD

[2]) Eingang MODE des FBs OUTPUT

© ifm electronic gmbh | www.ifm.com

13.2 Systemmerker

8374

Für die Programmierung sollten nur die Symbolnamen genutzt werden, da sich die zugehörigen Merkeradressen bei einer Erweiterung der Steuerungskonfiguration ändern können.

(→ Kapitel *Fehlercodes und Diagnoseinformationen* (→ Seite 48))

Systemmerker	Art	Beschreibung
TEMPERATURE	INT	Temperatur im Gerät [°C]
SUPPLY_VOLTAGE_VBBS	WORD	Versorgungsspannung an VBBs in [mV]
SUPPLY_VOLTAGE_VBBx	WORD	Versorgungsspannung an VBBx in [mV] CR040n: x = 1, 2
SUPPLY_VOLTAGE_VU	WORD	interne Versorgungsspannung

© ifm electronic gmbh | www.ifm.com

13.3 CANopen-Tabellen

Inhalt

IDs (Adressen) in CANopen 265
Aufbau von CANopen-Meldungen 266
Bootup-Nachricht 271
Netzwerk-Management (NMT) 272
CANopen Error-Code 276

9941

Die folgenden Tabellen informieren Sie über wichtige Werte und Einstellungen der CANopen-Schnittstellen.

13.3.1 IDs (Adressen) in CANopen

3952

In CANopen werden diverse Arten von 'Adressen' (hier: IDs) unterschieden:

- COB-ID
  Der **C**ommunication-**Ob**ject-**Id**entifier adressiert die Nachricht (= das Kommunikationsobjekt) im Geräteverzeichnis. Ein Kommunikationsobjekt besteht aus einem oder mehreren CAN-Nachrichten mit bestimmten Aufgaben, z.B.:
  - PDO (**P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten),
  - SDO (**S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten),
  - Emergency (Nachrichten-Objekt mit Notfalldaten),
  - Time (Nachrichten-Objekt mit Zeitangaben) oder
  - Error Control (Nachrichten-Objekt mit Fehlermeldungen).
- CAN-ID
  Der **CAN-Id**entifier definiert netzwerkweit CAN-Nachrichten. Der CAN-ID ist Hauptbestandteil des Arbitration-Feldes eines CAN-Datenübertragungsblocks. Je niedriger der CAN-ID, desto höher die Priorität der Meldung.
- Download-ID
  Der Download-ID bezeichnet den Node-ID für Service-Kommunikation per SDO für den Programm-Download und das Debuggen.
- Node-ID
  Der **Node-Id**entifier ist ein eindeutiger Bezeichner für CANopen-Geräte (Devices) im CAN-Netzwerk. Der Node-ID ist auch Bestandteil einiger vordefinierter Verbindungssätze (→ ***Funktions-Code / Predefined Connectionset*** (→ Seite 268)).

Vergleich Download-ID vs. COB-ID:

Controller Programm-Download		CANopen	
Download-ID	**COB-ID SDO**	**Node-ID**	**COB-ID SDO**
1…127	TX: 580_{16} + Download-ID	1…127	TX: 580_{16} + Node-ID
	RX: 600_{16} + Download-ID		RX: 600_{16} + Node-ID

TX = Slave sendet an Master
RX = Slave empfängt von Master

© ifm electronic gmbh | www.ifm.com

13.3.2 Aufbau von CANopen-Meldungen

Inhalt

Aufbau des COB-ID 267
Funktions-Code / Predefined Connectionset 268
SDO-Kommando-Bytes 269
SDO-Abbruch-Code 270

9971

Eine CANopen-Meldung besteht aus dem COB-ID und bis zu 8 Bytes Daten:

COB-ID			DLC	Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Details erfahren Sie in den folgenden Kapiteln.

Beachten Sie die umgekehrte Byte-Reihenfolge!

Beispiele:

Wert [hex]	Datentyp	Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
12	BYTE	1	2	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1234	WORD	3	4	1	2	–	–	–	–	–	–	–	–	–	–	–	–
12345678	DWORD	7	8	5	6	3	4	1	2	–	–	–	–	–	–	–	–

© ifm electronic gmbh | www.ifm.com

Aufbau des COB-ID

9972

Der erste Teil einer Meldung ist der COB-ID. Aufbau des 11-Bit COB-ID:

Nibble 0				Nibble 1				Nibble 2			
11	10	9	8	7	6	5	4	3	2	1	0
--	**3**	**2**	**1**	**0**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--	**Funktions-Code**				**Node-ID**						

Der COB-ID besteht aus ***Funktions-Code / Predefined Connectionset*** (→ Seite 268) und Node-ID.

Beispiel:
Das Kommunikations-Objekt = TPDO1 (TX)
Die Knoten-Nummer des Geräts = $20_{16} = 32_{10}$

Berechnung:
Der Funktions-Code für das Kommunikations-Objekt TPDO1 = 3_{16}
Die Wertigkeit des Funktions-Code im 11-Bit-COB-ID = $3_{16} \times 80_{16} = 180_{16}$
Dazu die Knoten-Nummer (20_{16}) addieren ⇨ der COB-ID lautet: $1A0_{16}$

1				A				0			
3	2	1	0	3	2	1	0	3	2	1	0
0	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
--	$3_{16} = 3_{10}$				$20_{16} = 32_{10}$						

© ifm electronic gmbh | www.ifm.com

Funktions-Code / Predefined Connectionset

9966

Im "CANopen Predefined Connectionset" sind einige Funktions-Codes vorbelegt.

Wenn Sie das Predefined Connectionset verwenden, können Sie ein CANopen-Netzwerk von bis zu 127 Teilnehmern in Betrieb nehmen, ohne dass es zu einer doppelten Vergabe von COB-IDs käme.

Broadcast- oder Multicast-Nachrichten:

Kommunikations-Objekt	Funktions-Code [hex]	COB-ID [hex]	zugehörige Parameter-Objekte [hex]
NMT	0	000	
SYNC	1	080	1005, 1006, 1007, 1028
TIME	2	100	1012, 1013

Punkt-zu-Punkt-Nachrichten:

Kommunikations-Objekt	Funktions-Code [hex]	COB-ID [hex]	zugehörige Parameter-Objekte [hex]
EMERGENCY	1	080 + Node-ID	1014, 1015
TPDO1 (TX)	3	180 + Node-ID	1800
RPDO1 (RX)	4	200 + Node-ID	1400
TPDO2 (TX)	5	280 + Node-ID	1801
RPDO2 (RX)	6	300 + Node-ID	1401
TPDO3 (TX)	7	380 + Node-ID	1802
RPDO3 (RX)	8	400 + Node-ID	1402
TPDO4 (TX)	9	480 + Node-ID	1803
RPDO4 (RX)	A	500 + Node-ID	1403
Default SSDO (TX)	B	580 + Node-ID	1200
Default CSDO (RX)	C	600 + Node-ID	1280
NMT Error Control	E	700 + Node-ID	1016, 1017

TX = Slave sendet an Master
RX = Slave empfängt von Master

SSDO = Server-SDO
CSDO = Client-SDO

© ifm electronic gmbh | www.ifm.com

SDO-Kommando-Bytes

9968

Aufbau einer SDO-Nachricht:

COB-ID	DLC	Kommando	Index		Sub-Index	Daten *)			
XXX	8	Byte	Byte 0	Byte 1	Byte	Byte 0	Byte 1	Byte 2	Byte 3

*) abhängig von den zu transportierenden Daten

Beachten Sie die umgekehrte Byte-Reihenfolge!

Ein SDO-COB-ID setzt sich wie folgt zusammen:

CANopen	
Node-ID	**COB-ID SDO**
1…127	TX: 580_{16} + Node-ID
	RX: 600_{16} + Node-ID

TX = Slave sendet an Master
RX = Slave empfängt von Master

DLC (Data length code) bezeichnet die Anzahl der Daten-Bytes (bei SDO: DLC = 8).

SDO-Kommando-Bytes:

Kommando hex	dez	Nachricht	Datenlänge	Beschreibung
21	33	Anforderung	mehr als 4 Bytes	Daten an Slave senden
22	34	Anforderung	1…4 Bytes	Daten an Slave senden
23	35	Anforderung	4 Bytes	Daten an Slave senden
27	39	Anforderung	3 Bytes	Daten an Slave senden
2B	43	Anforderung	2 Bytes	Daten an Slave senden
2F	47	Anforderung	1 Byte	Daten an Slave senden
40	64	Anforderung	---	Daten von Slave anfordern
42	66	Antwort	1…4 Bytes	Daten von Slave an Master senden
43	67	Antwort	4 Bytes	Daten von Slave an Master senden
47	71	Antwort	3 Bytes	Daten von Slave an Master senden
4B	75	Antwort	2 Bytes	Daten von Slave an Master senden
4F	79	Antwort	1 Byte	Daten von Slave an Master senden
60	96	Antwort	---	Datentransfer in Ordnung: Empfangsbestätigung von Slave an Master senden
80	128	Antwort	4 Bytes	Datentransfer fehlgeschlagen: Abbruch-Nachricht von Slave an Master senden → Kapitel ***SDO-Abbruch-Code*** (→ Seite 270)

© ifm electronic gmbh | www.ifm.com

SDO-Abbruch-Code

9970

Der SDO-Abbruch-Code gehört NICHT zum Emergency-Telegramm!

Abbruch-Code [hex]	Beschreibung
0503 0000	toggle bit not alternated
0504 0000	SDO protocol timed out
0504 0001	client/server command specifier not valid or unknown
0504 0002	invalid block size (block mode only)
0504 0003	invalid sequence number (block mode only)
0504 0004	CRC error (block mode only)
0504 0005	out of memory
0601 0000	unsupported access to an object
0601 0001	attempt to read a write only object
0601 0002	attempt to write a read only object
0602 0000	object does not exist in the object dictionary
0604 0041	object cannot be mapped to the PDO
0604 0042	the number and length of the objects to be mapped would exceed PDO length
0604 0043	general parameter incompatibility reason
0604 0047	general internal incompatibility in the device
0606 0000	access failed due to an hardware error
0607 0010	data type does not match, length of service parameter does not match
0607 0012	data type does not match, length of service parameter too high
0607 0013	data type does not match, length of service parameter too low
0609 0011	sub-index does not exist
0609 0030	value range of parameter exceeded (only for write access)
0609 0031	value of parameter written too high
0609 0032	value of parameter written too low
0609 0036	maximum value is less than minimum value
0800 0000	general error
0800 0020	data cannot be transferred or stored to the application
0800 0021	data cannot be transferred or stored to the application because of local control
0800 0022	data cannot be transferred or stored to the application because of the present device state
0800 0023	object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)

© ifm electronic gmbh | www.ifm.com

13.3.3 Bootup-Nachricht

9961

Der CAN-Teilnehmer sendet nach dem Booten einmalig die Bootup-Nachricht:

	Byte 1	Byte 0
hex	700_{16} + Node-ID	NMT-Status
dez	$1\,792_{10}$ + Node-ID	NMT-Status

Somit ist der Teilnehmer im CAN-Netzwerk lauffähig.

Beispiel:
Der Node-ID des Teilnehmers ist $7D_{16} = 125_{10}$.
Dann lautet Byte 1 der Bootup-Nachricht: $77D_{16} = 1\,917_{10}$

Es gibt Geräte, die kein [700_{16} + Node-ID] senden können.
Diese Geräte senden stattdessen folgende Bootup-Nachricht und ohne Status:

hex	80_{16} + Node-ID
dez	128_{10} + Node-ID

© ifm electronic gmbh | www.ifm.com

13.3.4 Netzwerk-Management (NMT)

9974

Netzwerk-Management-Kommandos

9962

Mit folgenden Netzwerk-Management-Kommandos kann der Anwender den Betriebsmodus von einzelnen oder allen CAN-Teilnehmern beeinflussen. Muster:

Byte 1	Byte 2	Byte 2
COB-ID	Kommando	Node-ID

Node-ID = 00 ⇨ Kommando gilt zeitgleich für alle Knoten im Netz

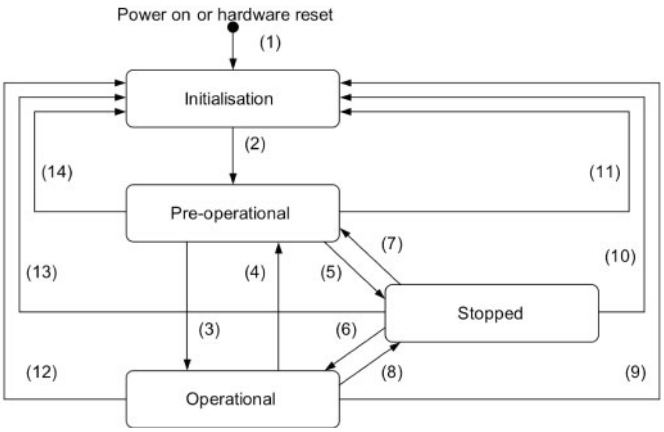
COB-ID	NMT-Kommando		Beschreibung	
00	$01_{16} = 01_{10}$	Node-ID	start_remode_node	CAN-Teilnehmer starten
00	$02_{16} = 02_{10}$	Node-ID	stop_remode_node	CAN-Teilnehmer stoppen
00	$80_{16} = 128_{10}$	Node-ID	enter_pre-operational	umschalten auf Pre-Operational
00	$81_{16} = 129_{10}$	Node-ID	reset_node	CAN-Teilnehmer zurücksetzen
00	$82_{16} = 130_{10}$	Node-ID	reset_communication	CAN-Kommunikation zurücksetzen

© ifm electronic gmbh | www.ifm.com

NMT-Status

9963

Das Status-Byte gibt Auskunft über den Zustand des CAN-Teilnehmers.



Grafik: Zustandsübergänge unter CANopen

Erlaubte Übergänge:

(1) Zustand wird bei Power On automatisch erreicht.

(2) interne Initialisierung ist beendet – Knoten geht automatisch nach PRE-OPERATIONAL

(3) NMT Service "Start Remote Node Indication"

(4) + (7) NMT Service "Enter PRE-OPERATIONAL Indication"

(5) + (8) NMT Service "Stop Remote Node Indication"

(6) NMT Service "Start Remote Node Indication"

(9)...(11) NMT Service "Reset Node Indication"

(12)...(14) NMT Service "Reset Communication Indication"

NMT-Status für CANopen-Master

9964

Status hex	dez	Beschreibung
00	0	nicht definiert
01	1	Master wartet auf die Bootup-Nachricht des Slaves. ODER: Master wartet auf Ablauf der GuardTime.
02	2	- Master wartet 300 ms. - Master fordert das Objekt 1000_{16} an. - Danach wechselt der Master auf Status 3.
03	3	Der Master konfiguriert seine Slaves. Dazu sendet der Master an die Slaves der Reihe nach alle vom Konfigurator erzeugten SDOs: - Der Master sendet an den Slave ein SDO-Read-Request (Index 1000_{16}). - Die generierten SDOs werden in ein SDO-Array gepackt. - Der Slave kennt seine erste SDO und die Anzahl seiner SDOs.
05	5	Nachdem an alle Slaves die SDOs übertragen wurden, geht der Master in den Status 5 und bleibt in diesem Status. Status 5 ist für den Master der normale Betriebszustand.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Ausgang NODE_STATE
CANOPEN_GETSTATE	Ausgang NODESTATE

© ifm electronic gmbh | www.ifm.com

NMT-Status für CANopen-Slave

9965

Status hex	dez	Beschreibung
FF	-1	Der Slave wird durch die NMT-Nachricht [Reset Node] zurückgesetzt und wechselt selbständig in den Status 1.
00	0	nicht definiert
01	1	Status = Warten auf BOOTUP Der Slave wechselt nach einer maximalen Zeit von 2 s oder sofort nach Empfang seiner Bootup-Message in den Status 2.
02	2	Status = BOOTUP Der Slave wechselt nach einer Verzögerungszeit von 0,5 s automatisch in den Status 3.
03	3	Status = PREPARED Im Status 3 wird der Slave konfiguriert. Der Slave bleibt solange im Status 3, bis er alle vom Konfigurator erzeugten SDOs erhalten hat. Dabei spielt es keine Rolle, ob während der Konfiguration vom Slave SDO-Transfers mit Abort (Fehler) oder ob alle fehlerfrei beantwortet wurden. Nur die vom Slave erhaltene Antwort als solche ist wichtig – nicht ihr Inhalt. Wenn im Konfigurator die Option [Knoten zurücksetzen] aktiviert wurde, wird nach dem Senden des Objekts 1011_{16} Subindex 1, der dann den Wert "load" enthält, ein erneuter Reset des Slaves durchgeführt. Der Slave wird dann wieder mit dem Upload des Objekts 1000_{16} angefragt. Slaves, bei denen während der Konfigurationsphase ein Problem auftritt, bleiben im Status 3 oder wechseln nach der Konfigurationsphase direkt in einen Fehlerstatus (Status > 5).
04	4	Status = PRE-OPERATIONAL Ein Knoten wechselt immer in den Status 4, außer: • es handelt sich um einen "optionalen" Slave und er wurde als nicht am Bus verfügbar detektiert (Abfrage Objekt 1000_{16}) ODER: • der Slave ist zwar vorhanden, aber hat auf die Abfrage des Objekts 1000_{16} mit einem anderen Typ in den unteren 16 Bits reagiert, als der Konfigurator erwartet hat.
05	5	Status = OPERATIONAL Im Status 5 findet der normale Datenaustausch statt: "Normal Operation". Wenn der Master auf [Automatisch starten] konfiguriert wurde, wird der Slave im Status 4 gestartet (d.h. es wird eine "Start Node"-NMT-Nachricht erzeugt) und der Slave wechselt automatisch nach Status 5. Wurde GLOBAL_START gesetzt, dann wird gewartet, bis sich alle Slaves im Status 4 befinden. Anschließend werden alle Slaves mit dem NMT-Kommando [Start All Nodes] gestartet.
61	97	Ein Knoten wechselt in den Status 97, wenn er optional ist (optionales Gerät in der CAN-Konfiguration) und nicht auf die SDO-Anfrage nach dem Objekt 1000_{16} reagiert hat. Wird der Slave zu einem späteren Zeitpunkt an das Netzwerk angeschlossen und erkannt, wird er automatisch gestartet. Dazu müssen Sie aber die Option [Automatisch starten] in den CAN-Parametern des Masters angewählt haben.
62	98	Ein Knoten wechselt in den Status 98, wenn der Gerätetyp (Objekt 1000_{16}) nicht dem konfigurierten Typ entspricht.
63	99	Im Falle eines Nodeguarding-Timeouts wird der Slave auf Status 99 gesetzt. Sobald der Slave wieder auf NodeGuard-Anfragen reagiert und die Option [Automatisch starten] eingeschaltet ist, wird er automatisch vom Master gestartet. Dabei wird der Knoten abhängig von seinem Status, der in der Antwort auf die Nodeguard-Anfragen enthalten ist, neu konfiguriert oder nur gestartet. Um den Slave manuell zu starten, genügt es, die Methode [NodeStart] zu benutzen.

Der Master sendet Nodeguard-Nachrichten an den Slave, ...
- wenn sich der Slave im Status 4 oder höher befindet UND
- wenn Nodeguarding konfiguriert wurde.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Ausgang NODE_STATE
CANOPEN_GETSTATE	Ausgang NODESTATE

© ifm electronic gmbh | www.ifm.com

CANopen-Status des Knotens

1973

Knotenstatus nach CANopen (mit diesen Werten wird der Status auch in den entsprechenden Nachrichten vom Knoten her codiert).

Status hex	dez	CANopen-Status	Beschreibung
00	0	BOOTUP	Knoten hat die BOOTUP-Nachricht erhalten.
04	4	PREPARED	Knoten wird per SDOs konfiguriert.
05	5	OPERATIONAL	Knoten nimmt am normalen Datenaustausch teil.
7F	127	PRE-OPERATIONAL	Knoten sendet keine Daten, ist aber vom Master konfigurierbar.

Wenn Nodeguarding aktiv: das höchstwertige Status-Bit wechselt (toggelt) von Nachricht zu Nachricht.

Knoten-Status aus FB lesen:

verwendeter Funktionsblock	hier steht dieser Knoten-Status
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Strukturelement LAST_STATE aus dem Array NODE_STATE_SLAVE
CANOPEN_GETSTATE	Ausgang LASTNODESTATE

© ifm electronic gmbh | www.ifm.com

13.3.5 CANopen Error-Code

Inhalt

Emergency-Nachrichten 276
Übersicht CANopen Error-Codes 277
Objekt 0x1001 (Error-Register) 278

9967

Emergency-Nachrichten

9973

Gerätefehler im Slave oder Probleme im CAN-Bus lösen Emergency-Nachrichten aus:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80_{16} + Node-ID		Error-Code		Objekt 1001_{16}	gerätespezifisch				

Beachten Sie die umgekehrte Byte-Reihenfolge!

© ifm electronic gmbh | www.ifm.com

Übersicht CANopen Error-Codes

8545

Error Code (hex)	Meaning / Bedeutung
00xx	Reset or no Error (Fehler rücksetzen / kein Fehler)
10xx	Generic Error (allgemeiner Fehler)
20xx	Current (Stromfehler)
21xx	Current, device input side (Stromfehler, eingangsseitig)
22xx	Current inside the device (Stromfehler im Geräteinnern)
23xx	Current, device output side (Stromfehler, ausgangsseitig)
30xx	Voltage (Spannungsfehler)
31xx	Mains Voltage
32xx	Voltage inside the device (Spannungsfehler im Geräteinnern)
33xx	Output Voltage (Spannungsfehler, ausgangsseitig)
40xx	Temperature (Temperaturfehler)
41xx	Ambient Temperature (Umgebungstemperaturfehler)
42xx	Device Temperature (Gerätetemperaturfehler)
50xx	Device Hardware (Geräte-Hardware-Fehler)
60xx	Device Software (Geräte-Software-Fehler)
61xx	Internal Software (Firmware-Fehler)
62xx	User Software (Applications-Software)
63xx	Data Set (Daten-/Parameterfehler)
70xx	Additional Modules (zusätzliche Module)
80xx	Monitoring (Überwachung)
81xx	Communication (Kommunikation)
8110	CAN Overrun-objects lost (CAN Überlauf-Datenverlust)
8120	CAN in Error Passiv Mode (CAN im Modus "fehlerpassiv")
8130	Life Guard Error or Heartbeat Error (Guarding-Fehler oder Heartbeat-Fehler)
8140	Recovered from Bus off (Bus-Off zurückgesetzt)
8150	Transmit COB-ID collision (Senden "Kollision des COB-ID")
82xx	Protocol Error (Protokollfehler)
8210	PDO not procedded due to length error (PDO nicht verarbeitet, fehlerhafte Längenangabe)
8220	PDO length exceeded (PDO Längenfehler, ausgangsseitig)
90xx	External Error (Externer Fehler)
F0xx	Additional Functions (zusätzliche Funktionen)
FFxx	Device specific (gerätespezifisch)

© ifm electronic gmbh | www.ifm.com

Objekt 0x1001 (Error-Register)

8547

Dieses Objekt spiegelt den allgemeinen Fehlerzustand eines CANopen-Gerätes wider. Das Gerät ist dann als fehlerfrei anzusehen, wenn das Objekt 1001_{16} keinen Fehler mehr signalisiert.

Bit	Meaning (Bedeutung)
0	Generic Error (allgemeiner Fehler)
1	Current (Stromfehler)
2	Voltage (Spannungsfehler)
3	Temperature (Temperaturfehler)
4	Communication Error (Kommunikationsfehler)
5	Device Profile specific (Geräteprofil spezifisch)
6	Reserved – always 0 (reserviert – immer 0)
7	manufacturer specific (herstellerspezifisch)

Für eine Fehlermeldung können mehrere Bits im Error-Register gleichzeitig gesetzt sein.

Beispiel: CR2033, Meldung "Leitungsbruch" an Kanal 2 (→ Installationsanleitung des Geräts):

COB-ID	DLC	Byte 0	Byte 1	Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
80_{16} + Node-ID		**00**	**FF**	**81**	**10**	**00**	00	00	00

Error-Code = $FF00_{16}$

Error-Register = 81_{16} = $1000\ 0001_2$, besteht also aus folgenden Fehlern:
- Generic Error (allgemeiner Fehler)
- manufacturer specific (herstellerspezifisch)

Betroffener Kanal = 0010_{16} = $0000\ 0000\ 0001\ 0000_2$ = Kanal 2

© ifm electronic gmbh | www.ifm.com

13.4 Arbeiten mit dem User-Flash-Speicher

Inhalt

Flash-Speicher – was ist das? 279
CSV-Datei – was ist das? 280
CSV-Datei erzeugen und übertragen 280

11607

Einige ifm-Geräte bieten einen User-Flash-Speicher an. Dies ist ein Flash-Speicher-Bereich, der dem Kunden für seine Applikationsdaten zur Verfügung steht.

Anwendungsbeispiele:
- Meldetexte (mehrsprachig umschaltbar) zur Anzeige in PDM und Display
- Belastungsgrenzwerte-Tabellen für z.B. Aufzüge, Krane und Drehleitern

Der Applikations-Programmierer erstellt dazu Listen oder Tabellen.
Das hierfür verwendete Programm muss die Quelldatei in eine CSV-Datei wandeln können. Geeignet sind z.B. Tabellenkalkulationsprogramme wie Microsoft Excel oder OpenOffice Calc.

13.4.1 Flash-Speicher – was ist das?

11608

Flash-ROM (oder Flash-EPROM oder Flash-Memory) kombiniert die Vorteile von Halbleiterspeicher und Festplatten. Wie jeder andere Halbleiterspeicher kommt Flash-Speicher ohne bewegliche Teile aus. Und die Daten bleiben wie bei einer Festplatte auch nach dem Ausschalten erhalten.

Der Flash-ROM hat sich aus dem EEPROM (**E**lectrical **E**rasable and **P**rogrammable **R**ead-**O**nly **M**emory) entwickelt. Beim Flash-ROM ist die Speicherung von Daten funktionell identisch wie beim EEPROM. Die Daten werden allerdings wie bei einer Festplatte blockweise in Datenblöcken zu 64, 128, 256, 1024, ... Byte zugleich geschrieben und gelöscht.

Vorteile von Flash-Speicher

- Die gespeicherten Daten bleiben auch bei fehlender Versorgungsspannung erhalten.
- Wegen fehlender beweglicher Teile ist Flash geräuschlos, unempfindlich gegen Erschütterungen und magnetische Felder.
- Im Vergleich zu Festplatten haben Flash-Speicher eine sehr kurze Zugriffszeit. Lese- und Schreibgeschwindigkeit sind über den gesamten Speicherbereich weitestgehend konstant.
- Die erreichbare Speichergröße ist durch die einfache und platzsparende Anordnung der Speicherzellen nach oben offen.

Nachteile von Flash-Speicher

- Begrenzte Zahl von Schreib- bzw. Löschvorgängen, die eine Speicherzelle vertragen kann:
  - Multi-Level-Cells: typ. 10 000 Zyklen
  - Single-Level-Cells: typ. 100 000 Zyklen
- Da ein Schreibvorgang Speicherblöcke zwischen 16 und 128 kByte gleichzeitig beschreibt, werden auch Speicherzellen beansprucht, die gar keiner Veränderung bedürfen.

©ifm electronic gmbh / www.ifm.com

13.4.2 CSV-Datei – was ist das?

11627

CSV = **C**omma **S**eparated **V**alues (auch: **C**haracter **S**eparated **V**alues)
Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.
• Die Dateinamen-Erweiterung lautet `.csv`.
• CoDeSys erwartet als Trennzeichen zwischen den Spalten der Quell-Tabelle ein Semikolon (;).
• CoDeSys erzeugt als Endezeichen einer Zeichenkette (String) ein Null-Byte (NUL).
• Jeder Datensatz (jede zu übertragende Tabellenzeile) sollte die gleiche Anzahl von Tabellenspalten haben.

Beispiel: Quell-Tabelle mit Zahlenwerten:

Wert 1.0	Wert 1.1	Wert 1.2	Wert 1.3
Wert 2.0	Wert 2.1	Wert 2.2	Wert 2.3
Wert 3.0	Wert 3.1	Wert 3.2	Wert 3.3

Daraus entsteht folgende CSV-Datei:

```
Wert 1.0;Wert 1.1;Wert 1.2;Wert 1.3
Wert 2.0;Wert 2.1;Wert 2.2;Wert 2.3
Wert 3.0;Wert 3.1;Wert 3.2;Wert 3.3
```

13.4.3 CSV-Datei erzeugen und übertragen

Inhalt

CSV-Datei und der ifm-Downloader 281
CSV-Datei und das ifm-Maintenance-Tool 288

11628

Abhängig vom ecomat*mobile*-Gerät, dessen Applikations-Programm mit einer CSV-Datei arbeiten soll, muss die CSV-Datei unterschiedliche Voraussetzungen erfüllen.

In folgender Tabelle sehen Sie, mit welchem Programm Sie arbeiten müssen, um die CSV-Datei in das ecomat*mobile*-Gerät zu übertragen:

Programm:	ifm-Downloader	ifm-Maintenance-Tool
Für folgende Geräte verfügbar: (hier: nur Geräte mit Flash-Speicher)	• CabinetController: CR030n • ClassicController: CR0020, CR0505 • ClassicController: CR0032, CR0033 • ExtendedController: CR0200 • ExtendedController: CR0232, CR0233 • Platinensteuerung: CS0015 • SafetyController: CR7021, CR7201, CR7506 • SmartController: CR25nn • PDM360smart: CR1070, CR1071	• BasicController: CR040n • BasicDisplay: CR0451
weiter mit Kapitel →	*CSV-Datei und der ifm-Downloader* (→ Seite 281)	*CSV-Datei und das ifm-Maintenance-Tool* (→ Seite 288)

© ifm electronic gmbh | www.ifm.com

CSV-Datei und der ifm-Downloader

Inhalt

Voraussetzungen für die CSV-Datei 281
CSV-Datei erstellen mittels Tabellenkalkulationsprogramm 282
CSV-Datei erstellen mittels Editor 284
CSV-Datei mit Downloader übertragen 286

11618

Folgende Geräte können mit dem ifm-Downloader kommunizieren:
- CabinetController: CR030n
- ClassicController: CR0020, CR0505
- ClassicController: CR0032, CR0033
- ExtendedController: CR0200
- ExtendedController: CR0232, CR0233
- Platinensteuerung: CS0015
- SafetyController: CR7021, CR7201, CR7506
- SmartController: CR25nn
- PDM360smart: CR1070, CR1071

Voraussetzungen für die CSV-Datei

11629

► Die Zeile für die Startadresse muss mit "ifm data at " beginnen!
Davor dürfen beliebige Zeilen stehen.

► Direkt danach müssen lückenlos die Daten folgen!
Struktur:
relative Adresse;Datum oder Text;Datentyp
Beispiel:
31;Text 02.....;string

► Alle Zeichenketten müssen die gleiche Länge haben!
Der ifm-Downloader kann nicht selbst für gleichlange Strings sorgen.

► Als Feld-Trenner gilt das Semikolon (;).

© ifm electronic gmbh | www.ifm.com

CSV-Datei erstellen mittels Tabellenkalkulationsprogramm

11633

Ein Skript ergänzt hier im Tabellenkalkulations-Programm automatisch den in Spalte D eingegebenen Text auf 20 Zeichen mittels angefügter Punkte oder schneidet nach dem 20. Zeichen den eingegebenen Text ab (→ Zeile 10).

	A	B	C	D
1	**Text to FLASH CR107n**			
2	Display text length must be = 20 characters			
3	If text is shorter: here fill with points (.) to illustrate this			
4				
5				
6	Start_Adr	Display text	typ	Input text
7	ifm data at 000F0000			
8	0	This is OK.....	string	This is OK
9	21	Flashtext 2.....	string	Flashtext 2
10	42	Text is longer than	string	Text is longer than 20 char.
11	63	Flashtext 4.....	string	Flashtext 4
12	84	Flashtext 5.....	string	Flashtext 5
13	105	Flashtext 6.....	string	Flashtext 6
14	126	Flashtext 7.....	string	Flashtext 7
15	147	Flashtext 8.....	string	Flashtext 8
16	168	Flashtext 9.....	string	Flashtext 9
17	189	Flashtext 10.....	string	Flashtext 10
18	210	Flashtext 11.....	string	Flashtext 11
19	231	Flashtext 12.....	string	Flashtext 12
20	252	Flashtext 13.....	string	Flashtext 13
21	273	Flashtext 14.....	string	Flashtext 14
22	294	Flashtext 15.....	string	Flashtext 15
23	315	Flashtext 16.....	string	Flashtext 16
24	336	Flashtext 17.....	string	Flashtext 17
25	357	Flashtext 18.....	string	Flashtext 18
26	378	Flashtext 19.....	string	Flashtext 19
27	399	Flashtext 20.....	string	Flashtext 20
28	420	Flashtext 21	string	Flashtext 21

Für das PDM360smart: CR1070, CR1071 gilt zusätzlich:
Alle Einträge zusammen müssen exakt 16 384 Bytes umfassen.
Dies wird möglich durch eine entsprechende Anzahl von Reserve-Zeilen:

780	16212	reserve.....	string	reserve
781	16233	reserve.....	string	reserve
782	16254	reserve.....	string	reserve
783	16275	reserve.....	string	reserve
784	16296	reserve.....	string	reserve
785	16317	reserve.....	string	reserve
786	16338	reserve.....	string	reserve
787	16359	reserve.....	string	reserve
788	16380	abcd	string	fill the file to 16 kBytes
789				

© ifm electronic gmbh www.ifm.com

Legende:

Feldnr.	Beschreibung
Zeilen 1...6	Beliebig als Kommentar, Überschrift, usw. nutzbar Für den Downloader ist kein besonderer Header erforderlich.
hier: A7	Startadresse (als Hex-Wert) des Flash-Speichers im Gerät (die Zeile für diesen Eintrag ist beliebig) Die konkrete Adresse mit dem Downloader ermitteln! Hier: Für CR107n startet der User Flash Speicher an der Adresse 000F 0000_{16} Die Zeile für die Startadresse muss mit "ifm data at " beginnen! In den Folgezeilen müssen lückenlos die Daten folgen!
A8...A787	Byte-Nummer für Beginn eines Meldetextes (oder einer Reserve) • Beginn mit der relativen Adresse 0 • folgende Adressen in Schritten von (Zeilenlänge plus 1) Bytes
B8...B787	Meldetexte (oder Reserve) • jede Zeile ist exakt 20 Zeichen lang • nur diese Daten werden nachher ins Gerät übertragen
C8...C787	Datentyp, hier: String(20), wobei für den Downloader die Länge nicht angegeben werden muss.
D8...D787	Felder zum Eintragen der Texte • Ein kleines Skript überträgt automatisch die Einträge in die Spalte B • das Skript passt dabei die Textlänge in Spalte B an • hier: Auffüllen mit Punkten
Zeile 788	Restliche Bytes zum Auffüllen des Flash-Speichers auf genau 16 kBytes • diese Zeile hat keine Verwendung

Diese Struktur ist erforderlich, damit die daraus erzeugte CSV-Datei von CoDeSys verstanden wird.

► Speichern Sie die Tabelle:
Speicherort wählen und Dateiname eintragen.
Wählen Sie dabei einen sinnvollen Dateinamen, um später die richtige Datei identifizieren zu können.

► Wandel Sie die Tabellen-Datei in eine CSV-Datei.
Wählen Sie das Semikolon ';' als Spaltentrenner.

Bei **Excel**: [Speichern unter...] > [Dateityp] = CSV-Datei

Bei **OpenOffice**: [Speichern unter...] > [Dateityp] = Text CSV > [Format beibehalten]
im Fenster [Textexport] einstellen:
• Semikolon als Feldtrenner
• Texttrenner = (leer)

► Warnmeldungen (wegen Formatierungsverlust) quittieren.

► Schließen Sie das Tabellenkalkulations-Programm.

© ifm electronic gmbh | www.ifm.com

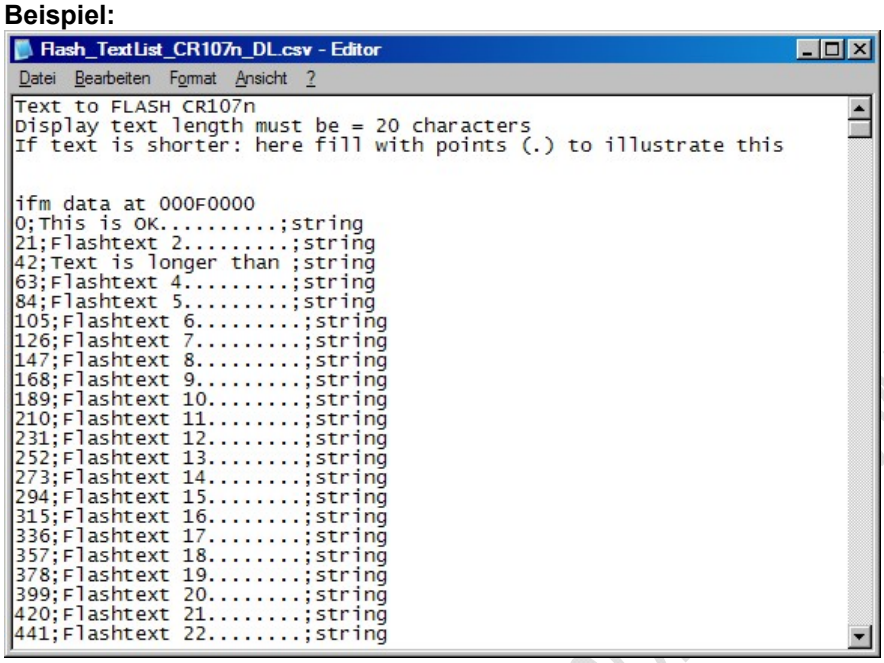
CSV-Datei erstellen mittels Editor

11634

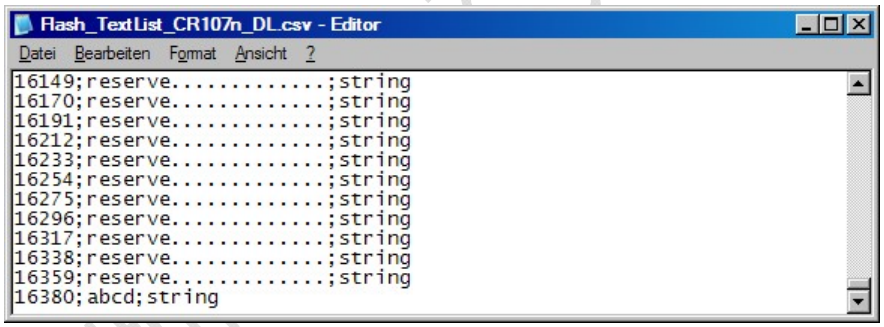
► Gewünschten Inhalt der CSV-Datei manuell eintragen.

► Datei als CSV-Datei speichern.
Dateityp = ANSI

Beispiel:



Für das PDM360smart: CR1070, CR1071 gilt zusätzlich:
Alle Einträge zusammen müssen exakt 16 384 Bytes umfassen.
Dies wird möglich durch eine entsprechende Anzahl von Reserve-Zeilen:



© ifm electronic gmbh www.ifm.com

Legende:

Eintrag	Beschreibung
Zeilen 1...4	Beliebig als Kommentar, Überschrift, usw. nutzbar Für den Downloader ist kein besonderer Header erforderlich.
Zeile 5	Startadresse (als Hex-Wert) des Flash-Speichers im Gerät (die Zeile für diesen Eintrag ist beliebig) Die konkrete Adresse mit dem Downloader ermitteln! Hier: Für CR107n startet der User Flash Speicher an der Adresse 000F 0000_{16} Die Zeile für die Startadresse muss mit "ifm data at " beginnen! In den Folgezeilen müssen lückenlos die Daten folgen!
0; 21; 42; ...	Byte-Nummer für Beginn eines Meldetextes (oder einer Reserve) • Beginn mit der relativen Adresse 0 • folgende Adressen in Schritten von (Zeilenlänge plus 1) Bytes
;Text;	Meldetexte (oder Reserve) • jede Zeile ist exakt 20 Zeichen lang (mit beliebigen Zeichen auffüllen, hier: Punkte) • der Datentyp ergibt sich automatisch aus der Zeilenlänge • nur diese Daten werden nachher ins Gerät übertragen
string	Datentyp, hier: String(20) • der Datentyp ergibt sich automatisch aus der Zeilenlänge
letzte Zeile	Restliche Bytes zum Auffüllen des Flash-Speichers auf genau 16 kBytes • diese Zeile hat keine Verwendung

© ifm electronic gmbh | www.ifm.com

CSV-Datei mit Downloader übertragen

11635

► Verbinden Sie die Programmierschnittstelle des ifm-Geräts mit der seriellen Schnittstelle oder mit der USB-Schnittstelle des PCs.

► Wenn noch nicht erfolgt, dann mit CoDeSys das Applikations-Programm (als Bootprojekt) in das ifm-Gerät übertragen.

► Im Downloader das Menü wählen [Options] > [File Options].
• Im Fenster [File Options] nur [User Flash] aktivieren, alle anderen Optionen deaktivieren.
• Im Feld CSV das Semikolon (;) als Spaltentrennzeichen eintragen.
• Mit [OK] abschließen.

► Die CSV-Datei in das ifm-Gerät übertragen:

Beispiel: PDM360smart

11623

ACHTUNG

Der Treiberbaustein der seriellen Schnittstelle kann beschädigt werden!

Beim Trennen oder Verbinden der seriellen Schnittstelle unter Spannung kann es zu undefinierten Zuständen kommen, die zu einer Schädigung des Treiberbausteins führen.

► Die serielle Schnittstelle nur im spannungslosen Zustand trennen oder verbinden!

► Das Gerät am PC anschließen.

► Den ifm-Downloader starten.

► Im Menü [Interface] die für Gerät und Anschluss geeignete Schnittstelle wählen.
Hier: RS232

► Schnittstellenparameter einstellen, z.B.:
• Baudrate [Bits per Second] = z.B. 57 600
• Databits = 8
• Parity = N
• Stop Bits = 1
• COM Port = z.B. COM1 (abhängig vom PC)

► Im Menü [Protocol] das CoDeSys-Protokoll wählen:

[Protocol] > [CoDeSys V2.3 C167]	Kommunikation über CoDeSys V2.3 C167 (Voreinstellung) Dies gilt für die folgenden Geräte: • CabinetController: CR030n • ClassicController: CR0020, CR0505 (RTS < V05) • ExtendedController: CR0200 (RTS < V05) • Platinensteuerung: CS0015 • SafetyController: CR7020, CR7200, CR7505 (RTS < V05) • SmartController: CR25nn • PDM360smart: CR1070, CR1071 (RTS < V05)
[Protocol] > [CoDeSys V2.3 Tricore]	Kommunikation über CoDeSys V2.3 Tricore Dies gilt für die folgenden Geräte: • ClassicController: CR0020, CR0505 (RTS ≥ V05) • ExtendedController: CR0200 (RTS ≥ V05) • ClassicController: CR0032, CR0033 • ExtendedController: CR0232, CR0233 • SafetyController: CR7020, CR7200, CR7505 (RTS ≥ V05) • SafetyController: CR7021, CR7201, CR7506 • PDM360smart: CR1070, CR1071 (RTS ≥ V05)

© ifm electronic gmbh | www.ifm.com

► Mit [Identity] prüfen:
• Funktioniert die Verbindung zum Gerät?
• Mit dem richtigen Gerät verbunden?

> Die in [File Options] gewählte Datei in das Gerät übertragen mit [Download].

> In einem Dialogfenster fragt die Software nach Speicherort im PC und Dateiname.

► Verzeichnis und Datei wählen.

► Mit [Öffnen] den Kopiervorgang starten.

> Wenn die CSV-Datei nicht in Ordnung ist:
Eine Fehlermeldung erscheint.

> Wenn die CSV-Datei richtig eingestellt ist und die richtige Länge hat:

> Wenn dies nicht bereits zuvor erfolgt war:
Das Gerät geht in den STOP-Modus.
Die LED leuchtet grün.

> Der Kopierprozess beginnt.
Ein Fortschrittsbalken zeigt an, wie weit der Vorgang bereits abgeschlossen ist.

> Abschließend erscheint eine Fertigmeldung.

► Mit [RUN] die Applikation auf dem Gerät starten.

> Die Applikation kann nun z.B. mit FLASH_INFO und FLASH_READ auf die Daten zugreifen.

© ifm electronic gmbh | www.ifm.com

CSV-Datei und das ifm-Maintenance-Tool

Inhalt

Voraussetzungen für die CSV-Datei 288
CSV-Datei erstellen mittels Tabellenkalkulationsprogramm 289
CSV-Datei erstellen mittels Editor 291
CSV-Datei mit Maintenance-Tool übertragen 293

11619

Folgende Geräte können mit dem ifm-Maintenance-Tool kommunizieren:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360: CR1050, CR1051

Voraussetzungen für die CSV-Datei

11630

► Die CSV-Datei muss eine bestimmte Kopfdaten-Struktur haben.
Alle Kopfdaten beginnen mit '#'.

- 1. Zeile: CSV-Datei-Typ
  z.B.: `#File Type=0`
  zulässig: 0/1
- 2. Zeile (optional): Projektname der CSV-Datei
  z.B.: `#Name=Demo Textmessages`
  zulässig: 0...20 Zeichen
- 3. Zeile (optional): Version der CSV-Datei
  z.B.: `#Version=V01.00.00`
  zulässig: 0...12 Zeichen

- Das ifm-Maintenance-Tool kennt selbst die Startadresse des User-Flash-Speichers.
  Die Adresse muss daher nicht in der CSV-Datei stehen.

► Direkt nach den Header-Zeilen müssen lückenlos die Daten folgen!

Struktur:
relative Adresse;Datum oder Text;Datentyp;{Kommentar}
Beispiel:
31;Übertemperatur;string(20);Text 02
Nach dem Datentyp MUSS ein Semikolon (;) folgen!

- Das ifm-Maintenance-Tool erzeugt selbst aus dem Datentyp die richtige Datenlänge.
  Die String-Daten müssen daher nicht mit der vollen Länge in der CSV-Datei stehen.

► Als Feld-Trenner gilt das Semikolon (;).

© ifm electronic gmbh | www.ifm.com

CSV-Datei erstellen mittels Tabellenkalkulationsprogramm

11639

Beispiel:

- Zweisprachige Meldetexte für BasicDisplay CR0451
- 20 Texte je bis zu 30 Zeichen in Deutsch
- 20 Texte je bis zu 30 Zeichen in Englisch

DemoFlashTextMessageD_E.xls

	A	B	C	D
1	#FileType=0			
2	#Name=Messages D/E CR0451			
3	#Version=V01.00.00			
4	0	Dies ist Text 1	String(30)	Text 01
5	31	Dies ist Text 2	String(30)	Text 02
6	62	Dies ist Text 3	String(30)	Text 03
7	93	Dies ist Text 4	String(30)	Text 04
8	124	Dies ist Text 5	String(30)	Text 05
9	155	Dies ist Text 6	String(30)	Text 06
10	186	Dies ist Text 7	String(30)	Text 07
11	217	Dies ist Text 8	String(30)	Text 08
12	248	Dies ist Text 9	String(30)	Text 09
13	279	Dies ist Text 10	String(30)	Text 10
14	310	Dies ist Text 11	String(30)	Text 11
15	341	Dies ist Text 12	String(30)	Text 12
16	372	Dies ist Text 13	String(30)	Text 13
17	403	Dies ist Text 14	String(30)	Text 14
18	434	Dies ist Text 15	String(30)	Text 15
19	465	Reserve	String(30)	Text 16
20	496	Reserve	String(30)	Text 17
21	527	Reserve	String(30)	Text 18
22	558	Reserve	String(30)	Text 19
23	589	Reserve	String(30)	Text 20
24	620	This is text 1	String(30)	Text 21
25	651	This is text 2	String(30)	Text 22
26	682	This is text 3	String(30)	Text 23
27	713	This is text 4	String(30)	Text 24
28	744	This is text 5	String(30)	Text 25
29	775	This is text 6	String(30)	Text 26
30	806	This is text 7	String(30)	Text 27
31	837	This is text 8	String(30)	Text 28
32	868	This is text 9	String(30)	Text 29
33	899	This is text 10	String(30)	Text 30
34	930	This is text 11	String(30)	Text 31
35	961	This is text 12	String(30)	Text 32
36	992	This is text 13	String(30)	Text 33
37	1023	This is text 14	String(30)	Text 34
38	1054	This is text 15	String(30)	Text 35
39	1085	reserve	String(30)	Text 36
40	1116	reserve	String(30)	Text 37
41	1147	reserve	String(30)	Text 38
42	1178	reserve	String(30)	Text 39
43	1209	reserve	String(30)	Text 40

Legende:

Feldnr.	Beschreibung
A1...A3	Tabellenkopf (Header); Einträge beginnen mit '#'
A1	#FileType= 0: Beim Übersetzten einer CSV-Datei werden die erfassten Parameter direkt in der Reihenfolge im User-Flash abgelegt 1: Beim Übersetzten einer CSV-Datei werden die erfassten Parameter so im User-Flash abgelegt, dass die Daten mittels CoDeSys-Struktur direkt ausgelesen werden können.

(c) | www.ifm.com

Feldnr.	Beschreibung
A2	#Name= Name zum Definieren der Tabelle und zum Finden der Tabelle im Applikations-Programm Länge = 0...20 Zeichen
A3	#Version= Version der Tabelle (z.B. für verschiedene Fahrzeuge) Länge = 0...12 Zeichen
A4...A43	Byte-Nummer für Beginn eines Meldetextes
A4	erster Text der ersten Sprache (Byte-Nummer = 0)
A24	hier: erster Text der zweiten Sprache (Byte-Nummer = 620 = Offset)
B4...B23 B24...B43	Meldetexte (erste Sprache) die selben Meldetexte (zweite Sprache) Nur diese Texte überträgt das ifm-Maintenance-Tool in das Gerät
C4...C43	Datentyp, hier: String(30)
D4...D43	Kommentare (optional) • nur zur Information beim Erstellen der Tabelle • Kommentare werden nicht in das Gerät übertragen

► Diese Struktur ist erforderlich, damit die daraus erzeugte CSV-Datei von CoDeSys verstanden wird.

► Speichern Sie die Tabelle:
Speicherort wählen und Dateiname eintragen.
Wählen Sie dabei einen sinnvollen Dateinamen, um später die richtige Datei identifizieren zu können.

► Wandel Sie die Tabellen-Datei in eine CSV-Datei.
Wählen Sie das Semikolon ';' als Spaltentrenner.

Bei **Excel**: [Speichern unter...] > [Dateityp] = CSV-Datei

Bei **OpenOffice**: [Speichern unter...] > [Dateityp] = Text CSV > [Format beibehalten]
im Fenster [Textexport] einstellen:
• Semikolon als Feldtrenner
• Texttrenner = (leer)

► Warnmeldungen (wegen Formatierungsverlust) quittieren.

► Schließen Sie das Tabellenkalkulations-Programm.

► Die erzeugte CSV-Datei mit einem Editor öffnen:

```
#FileType=0;;;
#Name=Messages D/E CR0451;;;
#version=V01.00.00;;;
0;Dies ist Text 1;String(30);Text 01
31;Dies ist Text 2;String(30);Text 02
62;Dies ist Text 3;String(30);Text 03
93;Dies ist Text 4;String(30);Text 04
124;Dies ist Text 5;String(30);Text 05
155;Dies ist Text 6;String(30);Text 06
186;Dies ist Text 7;String(30);Text 07
217;Dies ist Text 8;String(30);Text 08
248;Dies ist Text 9;String(30);Text 09
279;Dies ist Text 10;String(30);Text 10
```

► Hinter den Header-Zeilen (beginnen mit '#') alle Semikola entfernen.

► Schließen Sie den Editor mit Speichern der Datei.

ifm electronic gmbh | www.ifm.com

CSV-Datei erstellen mittels Editor

11640

► Gewünschten Inhalt der CSV-Datei manuell eintragen.

► Datei als CSV-Datei speichern.
Dateityp = ANSI

Beispiel:
- Zweisprachige Meldetexte für BasicDisplay CR0451
- 20 Texte je bis zu 30 Zeichen in Deutsch
- 20 Texte je bis zu 30 Zeichen in Englisch

```
#FileType=0
#Name=Messages D/E CR0451
#Version=V01.00.00
0;Dies ist Text 1;String(30);
31;Dies ist Text 2;String(30);
62;Dies ist Text 3;String(30);
93;Dies ist Text 4;String(30);
124;Dies ist Text 5;String(30);
155;Dies ist Text 6;String(30);
186;Dies ist Text 7;String(30);
217;Dies ist Text 8;String(30);
248;Dies ist Text 9;String(30);
279;Dies ist Text 10;String(30);
310;Dies ist Text 11;String(30);
341;Dies ist Text 12;String(30);
372;Dies ist Text 13;String(30);
403;Dies ist Text 14;String(30);
434;Dies ist Text 15;String(30);
465;Reserve;String(30);
496;Reserve;String(30);
527;Reserve;String(30);
558;Reserve;String(30);
589;Reserve;String(30);
620;This is text 1;String(30);
651;This is text 2;String(30);
682;This is text 3;String(30);
713;This is text 4;String(30);
744;This is text 5;String(30);
775;This is text 6;String(30);
806;This is text 7;String(30);
837;This is text 8;String(30);
868;This is text 9;String(30);
899;This is text 10;String(30);
930;This is text 11;String(30);
961;This is text 12;String(30);
992;This is text 13;String(30);
1023;This is text 14;String(30);
1054;This is text 15;String(30);
1085;reserve;String(30);
1116;reserve;String(30);
1147;reserve;String(30);
1178;reserve;String(30);
1209;reserve;String(30);
```

© ifm electronic gmbh | www.ifm.com

Legende:

Feldnr.	Beschreibung
Zeilen 1...3	Tabellenkopf (Header); Einträge beginnen mit '#'
Zeile 1	#FileType= 0: Beim Übersetzten einer CSV-Datei werden die erfassten Parameter direkt in der Reihenfolge im User-Flash abgelegt 1: Beim Übersetzten einer CSV-Datei werden die erfassten Parameter so im User-Flash abgelegt, dass die Daten mittels CoDeSys-Struktur direkt ausgelesen werden können.
Zeile 2 (optional)	#Name= Name zum Definieren der Tabelle und zum Finden der Tabelle im Applikations-Programm Länge = 0...20 Zeichen
Zeile 3 (optional)	#Version= Version der Tabelle (z.B. für verschiedene Fahrzeuge) Länge = 0...12 Zeichen
0; 31; 62; ...	Byte-Nummer für Beginn eines Meldetextes • Beginn mit der relativen Adresse 0 • folgende Adressen in Schritten von (Zeilenlänge plus 1) Bytes
;Text;	Meldetexte (oder Reserve) • jede Zeile ist exakt 20 Zeichen lang (mit beliebigen Zeichen auffüllen, hier: Punkte) • der Datentyp ergibt sich automatisch aus der Zeilenlänge • nur diese Daten werden nachher ins Gerät übertragen
String(30)	Datentyp Nach dem Datentyp MUSS ein Semikolon (;) folgen! Das ifm-Maintenance-Tool generiert aus der Angabe automatisch die richtige Länge der Meldetexte

© ifm electronic gmbh | www.ifm.com

CSV-Datei mit Maintenance-Tool übertragen

11641

► Verbinden Sie die Programmierschnittstelle des ifm-Geräts mit dem PC.

► Wenn noch nicht erfolgt, dann mit CoDeSys das Applikations-Programm (als Bootprojekt) in das ifm-Gerät übertragen.

► Das ifm-Maintenance-Tool starten.

Z.B. für das BasicSystem:

► Wenn noch nicht geschehen, folgendes Menü herstellen:
[ecomat mobile] > [CAN] > [Basic System]

► Im Maintenance-Tool in der linken Spalte der Programmoberfläche folgendes Menü wählen:
[ecomat mobile] > [CAN] > [Basic System]

► In der mittleren Spalte der Programmoberfläche folgendes Menü wählen:
[Basic System] > [System-Information] > [Identität]

> Nach Klick auf [Lesen] erscheint im rechten Teil der Programmoberfläche die Geräteinformation.

Wenn die Daten des richtigen Geräts erscheinen:

► In der mittleren Spalte der Programmoberfläche folgendes Menü wählen:
[Basic System] > [Software] > [Laden]

► Im rechten Teil der Programmoberfläche klicken auf [nach Basic System]

► Im Feld [Software laden] klicken auf [*.csv-Datei importieren...].

> Das Fenster [Software laden] erscheint.

► Speicherort und Datei wählen und mit [Öffnen] bestätigen.

> Ein Informationsfenster erscheint mit folgenden Angaben:
- Speicherort, -Pfad
- Speicher: n Bytes von m Bytes verwendet
- Datei-Typ
- Name (aus den CSV-Kopfdaten, editierbar)
- Version (aus den CSV-Kopfdaten, editierbar)

► Mit [Importieren] die Datei in die Liste der zu übertragenden Dateien aufnehmen.

► Die zu übertragenden Dateien (oder: alle) markieren.

► Mit [Laden] die CSV-Datei in das ifm-Gerät übertragen.

> Ein Fortschrittsbalken zeigt an, wie weit der Vorgang bereits abgeschlossen ist.

> Abschließend erscheint eine Fertigmeldung.

► Das Gerät neu starten.

> Die Applikation kann nun z.B. mit FLASH_INFO und FLASH_READ auf die Daten zugreifen.

© ifm electronic gmbh | www.ifm.com

13.5 Übersicht der verwendeten Dateien und Bibliotheken

Inhalt

Dateien und Bibliotheken im Gerät installieren 294
Allgemeine Übersicht 295
Wozu dienen die einzelnen Dateien und Bibliotheken? 297

2711

(Stand: 02.03.2011)

Je nach Gerät und gewünschter Funktion kommen verschiedene Bibliotheken und Dateien zum Einsatz. Teilweise werden sie automatisch geladen oder müssen vom Programmierer eingefügt oder geladen werden.

13.5.1 Dateien und Bibliotheken im Gerät installieren

2721

Werkseinstellung: Das Gerät enthält nur den Bootloader.

► Betriebssystem (`*.H86` oder `*.RESX`) laden.

► Projekt (`*.PRO`) im PC anlegen: Target (`*.TRG`) eintragen.

► Zusätzlich je nach Gerät und Target-Version:
Steuerungskonfiguration (`*.CFG`) festlegen.

> CoDeSys bindet die zum Target zugehörenden Dateien in das Projekt ein:
`*.TRG, *.CFG, *.CHM, *.INI, *.LIB`.

► Bei Bedarf das Projekt mit weiteren Bibliotheken (`*.LIB`) ergänzen.

Bestimmte Bibliotheken binden automatisch weitere Bibliotheken in das Projekt ein:
z.B. basieren einige Funktionsblöcke in ifm-Bibliotheken (`ifm_*.LIB`) auf Funktionsblöcken in CoDeSys-Bibliotheken (`3S_*.LIB`).

© ifm electronic gmbh | www.ifm.com

13.5.2 Allgemeine Übersicht

2712

Dateiname	Beschreibung und Speicherort [3])
ifm_CRnnnn_Vxxyyzz.CFG [1]) ifm_CRnnnn_Vxx.CFG [2])	Steuerungskonfiguration je Gerät nur 1 gerätespezifische Datei enthält: IEC- und symbolische Adressen der Ein- und Ausgänge, der Systemmerker sowie die Speicherverteilung …\CoDeSys V*\Targets\ifm\ifm_CRnnnncfg\Vxxyyzz
CAA-*.CHM	Online-Hilfe je Gerät nur 1 gerätespezifische Datei enthält: Online-Hilfe zu diesem Gerät …\CoDeSys V*\Targets\ifm\Help\… (Sprache)
ifm_CRnnnn_Vxxyyzz.H86 ifm_CRnnnn_Vxxyyzz.RESX	Betriebssystem / Laufzeitsystem (muss bei Erstbenutzung in den Controller / Monitor geladen werden) je Gerät nur 1 gerätespezifische Datei …\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn
ifm_Browser_CRnnnn.INI	CoDeSys-Bowser-Kommandos (CoDeSys benötigt die Datei zum Projektstart) je Gerät nur 1 gerätespezifische Datei enthält: Kommandos für Browser in CoDeSys …\CoDeSys V*\Targets\ifm
ifm_Errors_CRnnnn.INI	CoDeSys-Fehler-Datei (CoDeSys benötigt die Datei zum Projektstart) je Gerät nur 1 gerätespezifische Datei enthält: gerätespezifische Fehlermeldungen aus CoDeSys …\CoDeSys V*\Targets\ifm
ifm_CRnnnn_Vxx.TRG	Target-Datei je Gerät nur 1 gerätespezifische Datei enthält: Hardware-Beschreibung für CoDeSys, z.B.: Speicher, Dateiablageorte …\CoDeSys V*\Targets\ifm
ifm_*_Vxxyyzz.LIB	allgemeine Bibliotheken je Gerät mehrere Dateien möglich …\CoDeSys V*\Targets\ifm\Library
ifm_CRnnnn_Vxxyyzz.LIB	gerätespezifische Bibliothek je Gerät nur 1 gerätespezifische Datei enthält: Programmbausteine dieses Geräts …\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn
ifm_CRnnnn_*_Vxxyyzz.LIB	gerätespezifische Bibliotheken je Gerät mehrere Dateien möglich → folgende Tabellen …\CoDeSys V*\Targets\ifm\Library\ifm_CRnnnn

Legende:

*	beliebige Zeichen
CRnnnn	Artikelnummer des Controllers / Monitors
V*	CoDeSys-Version
Vxx	Versionsnummer der ifm-Software
yy	Release-Nummer der ifm-Software
zz	Patch-Nummer der ifm-Software

[1]) gültig für CRnn32 Target-Version bis V01, alle anderen Geräte bis V04

[2]) gültig für CRnn32 Target-Version ab V02, CR040n ab V01, alle anderen Geräte ab V05

[3]) Speicherort der Dateien:
System-Laufwerk (C: / D:) \ Programme-Ordner \ ifm electronic

© ifm electronic gmbh | www.ifm.com

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Betriebssystems (`ifm_CRnnnn_Vxxyyzz.H86` / `ifm_CRnnnn_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CRnnnn_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CRnnnn_Vxxyyzz.LIB`) und
- der weiteren Dateien (→ Kapitel ***Übersicht der verwendeten Dateien und Bibliotheken*** (→ Seite 279)).

CRnnnn	Geräte-Artikelnummer
Vxx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0032`") und die Software-Versionsnummer "xx" (z.B. "`02`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 1131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CoDeSys nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

© ifm electronic gmbh | www.ifm.com

13.5.3 Wozu dienen die einzelnen Dateien und Bibliotheken?

Inhalt

Dateien für Betriebssystem / Laufzeitsystem 297
Target-Datei 297
Steuerungskonfigurations-Datei 297
ifm-Gerätebibliotheken 297
ifm-CANopen-Hilfsbibliotheken Master/Slave 298
CoDeSys-CANopen-Bibliotheken 298
spezielle ifm-Bibliotheken 299

2713

Die nachfolgende Übersicht zeigt, welche Dateien/Bibliotheken mit welchem Gerät eingesetzt werden können und dürfen. Dateien/Bibliotheken, die in dieser Liste nicht aufgeführt werden, können nur unter bestimmten Bedingungen eingesetzt werden oder die Funktionalität wurde noch nicht getestet.

Dateien für Betriebssystem / Laufzeitsystem

2714

Dateiname	Funktion	verfügbar für:
ifm_CRnnnn_Vxxyyzz.H86 ifm_CRnnnn_Vxxyyzz.HEX	Betriebssystem / Laufzeitsystem	• alle ecomatmobile-Controller • BasicDisplay: CR0451 • PDM: CR10nn
ifm_Browser_CRnnnn.INI	CoDeSys-Browser-Kommandos	• alle ecomatmobile-Controller • PDM: CR10nn
ifm_Errors_CRnnnn.INI	CoDeSys-Fehler-Datei	• alle ecomatmobile-Controller • PDM: CR10nn

Target-Datei

2715

Dateiname	Funktion	verfügbar für:
ifm_CRnnnn_Vxx.TRG	Target-Datei	• alle ecomatmobile-Controller • BasicDisplay: CR0451 • PDM: CR10nn

Steuerungskonfigurations-Datei

2716

Dateiname	Funktion	verfügbar für:
ifm_CRnnnn_Vxxyyzz.CFG	Steuerungskonfiguration	• alle ecomatmobile-Controller • BasicDisplay: CR0451 • PDM: CR10nn

ifm-Gerätebibliotheken

2717

Dateiname	Funktion	verfügbar für:
ifm_CRnnnn_Vxxyyzz.LIB	gerätespezifische Bibliothek	• alle ecomatmobile-Controller • BasicDisplay: CR0451 • PDM: CR10nn
ifm_CR0200_MSTR_Vxxyyzz.LIB	Bibliothek ohne Extended-Funktionen	• ExtendedController: CR0200
ifm_CR0200_SMALL_Vxxyyzz.LIB	Bibliothek ohne Extended-Funktionen, reduzierter Funktionsumfang	• ExtendedController: CR0200

© ifm electronic gmbh | www.ifm.com

ifm-CANopen-Hilfsbibliotheken Master/Slave

2718

Diese Bibliotheken setzen auf CoDeSys-Bibliotheken (3S-CANopen-Funktionen) auf und stellen sie dem Anwender übersichtlich zur Verfügung.

Dateiname	Funktion	verfügbar für:
ifm_CRnnnn_CANopenMaster_Vxxyyzz.LIB	CANopen Master Emergency- und Status-Handler	• alle ecomatmobile-Controller *) • PDM: CR10nn *)
ifm_CRnnnn_CANopenSlave_Vxxyyzz.LIB	CANopen Slave Emergency- und Status-Handler	• alle ecomatmobile-Controller *) • PDM: CR10nn *)
ifm_CANx_SDO_Vxxyyzz.LIB	CANopen SDO Read und SDO Write	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_CANopen_NT_Vxxyyzz.LIB	CANopen-Bausteine im CAN-Stack	• BasicController: CR040n • BasicDisplay: CR0451 • PDM360NG: CR108n

*) jedoch NICHT für...
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

CoDeSys-CANopen-Bibliotheken

2719

Diese Bibliotheken sind für folgende Geräte NICHT verwendbar:
- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

Dateiname	Funktion	verfügbar für:
3S_CanDrvOptTable.LIB [1] 3S_CanDrvOptTableEx.LIB [2]	CANopen Treiber	• alle ecomatmobile-Controller • PDM360smart: CR1070, CR1071
3S_CanDrv.LIB [3]		• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
3S_CANopenDeviceOptTable.LIB [1] 3S_CANopenDeviceOptTableEx.LIB [2]	CANopen Slave-Treiber	• alle ecomatmobile-Controller • PDM360smart: CR1070, CR1071
3S_CANopenDevice.LIB [3]		• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
3S_CANopenManagerOptTable.LIB [1] 3S_CANopenManagerOptTableEx.LIB [2]	CANopen Netzwerkmanager	• alle ecomatmobile-Controller • PDM360smart: CR1070, CR1071
3S_CANopenManager.LIB [3]		• PDM360: CR1050, CR1051 • PDM360compact: CR1053, CR1056
3S_CANopenMasterOptTable.LIB [1] 3S_CANopenMasterOptTableEx.LIB [2]	CANopen Master	• alle ecomatmobile-Controller • PDM360smart: CR1070, CR1071
3S_CANopenMaster.LIB [3]		• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
3S_CANopenNetVarOptTable.LIB [1] 3S_CANopenNetVarOptTableEx.LIB [2]	Treiber für Netzwerkvariablen	• alle ecomatmobile-Controller • PDM360smart: CR1070, CR1071
3S_CANopenNetVar.LIB [3]		• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056

[1]) gültig für CRnn32 Target-Version bis V01, alle anderen Geräte bis V04
[2]) gültig für CRnn32 Target-Version ab V02, alle anderen Geräte ab V05
[3]) Für folgende Geräte gilt: diese Bibliothek ist funktionslos als Platzhalter enthalten:

© ifm electronic gmbh | www.ifm.com

- BasicController: CR040n
- BasicDisplay: CR0451
- PDM360NG: CR108n

spezielle ifm-Bibliotheken

2720

Dateiname	Funktion	verfügbar für:
ifm_RawCAN_NT_Vxxyyzz.LIB	CAN-Bausteine im CAN-Stack auf Basis Layer 2	• BasicController: CR040n • BasicDisplay: CR0451 • PDM360NG: CR108n
ifm_J1939_NT_Vxxyyzz.LIB	J1939 Kommunikationsfunktionen im CAN-Stack	• BasicController: CR040n • BasicDisplay: CR0451 • PDM360NG: CR108n
ifm_NetVarLib_NT_Vxxyyzz.LIB	zusätzlicher Treiber für Netzwerkvariablen	• BasicController: CR040n • BasicDisplay: CR0451 • PDM360NG: CR108n
ifm_J1939_Vxxyyzz.LIB	J1939 Kommunikationsfunktionen	bis Target V04 für: • CabinetController: CR0303 • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7020, CR7200, CR7505 • SmartController: CR2500 • PDM360smart: CR1070, CR1071
ifm_J1939_x_Vxxyyzz.LIB	J1939 Kommunikationsfunktionen	ab Target V05 für: • CabinetController: CR0303 • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7nnn • SmartController: CR2500 • PDM360smart: CR1070, CR1071
ifm_CRnnnn_J1939_Vxxyyzz.LIB	J1939 Kommunikationsfunktionen	• ClassicController: CR0032, CR0033 • ExtendedController: CR0232, CR0233
ifm_PDM_J1939_Vxxyyzz.LIB	J1939 Kommunikationsfunktionen	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_CANx_LAYER2_Vxxyyzz.LIB	CAN-Bausteine auf Basis Layer 2: CAN Transmit, CAN Receive	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_CAN1E_Vxxyyzz.LIB	Stellt den CAN-Bus von 11 Bit auf 29 Bit um	bis Target V04 für: • PDM360smart: CR1070, CR1071
ifm_CAN1_EXT_Vxxyyzz.LIB	Stellt den CAN-Bus von 11 Bit auf 29 Bit um	ab Target V05 für: • CabinetController: CR030n • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • Platinensteuerung: CS0015 • SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506 • SmartController: CR25nn • PDM360smart: CR1070, CR1071
ifm_CAMERA_O2M_Vxxyyzz.LIB	Kamera-Funktionen	• PDM360: CR1051
CR2013AnalogConverter.LIB	Analogwertkonvertierung für E/A-Modul CR2013	• alle ecomatmobile-Controller • PDM: CR10nn
ifm_Hydraulic_16bitOS04_Vxxyyzz.LIB	Hydraulikfunktionen für Controller	bis Target V04 für: • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7020, CR7200, CR7505 • SmartController: CR25nn

© ifm electronic gmbh | www.ifm.com

Dateiname	Funktion	verfügbar für:
ifm_Hydraulic_16bitOS05_Vxxyyzz.LIB	Hydraulikfunktionen für Controller	ab Target V05 für: • ClassicController: CR0020, CR0505 • ExtendedController: CR0200 • SafetyController: CR7020, CR7021, CR7200, CR7201, CR7505, CR7506 • SmartController: CR25nn
ifm_Hydraulic_32bit_Vxxyyzz.LIB	Hydraulikfunktionen für Controller	• ClassicController: CR0032, CR0033 • ExtendedController: CR0232, CR0233
ifm_Hydraulic_CR0303_Vxxyyzz.LIB	Hydraulikfunktionen für Controller	ab Target V05 für: • CabinetController: CR0303
ifm_SafetyIO_Vxxyyzz.LIB	Sicherheitsfunktionen	• SafetyController: CR7nnn
ifm_PDM_UTIL_Vxxyyzz.LIB	Hilfsfunktionen PDM	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_PDMng_UTIL_Vxxyyzz.LIB	Hilfsfunktionen PDM	• PDM360NG: CR108n
ifm_PDMsmart_UTIL_Vxxyyzz.LIB	Hilfsfunktionen PDM	• PDM360smart: CR1070, CR1071
ifm_PDM_Input_Vxxyyzz.LIB	alternative Eingabefunktionen PDM	• PDM: CR10nn
ifm_CR107n_Init_Vxxyyzz.LIB	Initialisierungsfunktion PDM360smart	• PDM360smart: CR1070, CR1071
ifm_PDM_File_Vxxyyzz.LIB	Dateifunktionen PDM360	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056 • PDM360NG: CR108n
ifm_PDM360NG_linux_syscall_asynch.LIB	Linux-Kommandos an das System senden	• PDM360NG: CR108n
ifm_PDM360NG_USB_Vxxyyzz.LIB	Geräte an der USB-Schnittstelle verwalten	• PDM360NG: CR108n
ifm_PDM360NG_USB_LL_Vxxyyzz.LIB	Hilfsbibliothek für ifm_PDM360NG_USB_Vxxyyzz.LIB	• PDM360NG: CR108n
Instrumente_x.LIB	vordefinierte Anzeige-Instrumente	• PDM: CR10nn
Symbols_x.LIB	vordefinierte Symbole	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056
Segment_x.LIB	vordefinierte 7-Segment-Anzeigen	• PDM360: CR1050, CR1051 • PDM360compact: CR1052, CR1053, CR1055, CR1056

Weitere Bibliotheken auf Anfrage.

© ifm electronic gmbh | www.ifm.com

14 Begriffe und Abkürzungen

Adresse

Das ist der „Name“ des Teilnehmers im Bus. Alle Teilnehmer benötigen eine unverwechselbare, eindeutige Adresse, damit der Austausch der Signale fehlerfrei funktioniert.

Anforderungsrate rd

Die Anforderungsrate r_d ist die Häufigkeit je Zeiteinheit von Anforderungen an eine sicherheitsgerichtete Reaktion eines SRP/CS.

Anleitung

Übergeordnetes Wort für einen der folgenden Begriffe:
Montageanleitung, Datenblatt, Benutzerinformation, Bedienungsanleitung, Gerätehandbuch, Installationsanleitung, Onlinehilfe, Systemhandbuch, Programmierhandbuch, usw.

Applikations-Software

Software, die speziell für die Applikation (Anwendung) vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

Für sicherheitsrelevante Teile von Steuerungen (→SRP/CS) müssen spezielle Anforderungen erfüllt sein.
→ Programmiersprache, sicherheitsrelevant

Architektur

Spezifische Konfiguration von Hardware- und Software-Elementen in einem System.

Ausfall

Ausfall ist die Beendigung der Fähigkeit einer Einheit, eine geforderte Funktion zu erfüllen.

Nach einem Ausfall hat die Einheit einen Fehler. Der Ausfall ist ein Ereignis, der Fehler ein Zustand.

Der so definierte Begriff kann nicht auf Einheiten angewendet werden, die nur aus Software bestehen.

Ausfall, gefahrbringend

Ein gefahrbringender Ausfall hat das Potential, das SRP/CS in einen gefährlichen Zustand oder eine Fehlfunktion zu bringen. Ob dieses Potential bemerkt werden kann oder nicht, hängt von der Architektur des Systems ab. In einem redundanten System wird ein gefährlicher Hardware-Ausfall weniger wahrscheinlich zu einem gefährlichen Ausfall des Gesamtsystems führen.

Ausfall, systematischer

Ein systematischer Ausfall ist ein Ausfall mit deterministischem (nicht zufälligem) Bezug zu einer bestimmten Ursache. Der systematische Ausfall kann nur beseitigt werden durch Änderung des Entwurfs oder des Herstellprozesses, Betriebsverfahren, Dokumentation oder zugehörigen Faktoren.

Eine Instandsetzung ohne Änderung des Systems wird den Grund des systematischen Ausfalls in der Regel nicht beseitigen.

Baud

Baud, Abk.: Bd = Maßeinheit für die Geschwindigkeit bei der Datenübertragung. Baud ist nicht zu verwechseln mit "bits per second" (bps, Bit/s). Baud gibt zwar die Anzahl von Zustandsänderungen (Schritte, Takte) pro Sekunde auf einer Übertragungsstrecke an. Aber es ist nicht festgelegt, wie viele Bits pro Schritt übertragen werden. Der Name Baud geht auf den französischen Erfinder J. M. Baudot zurück, dessen Code für Telexgeräte verwendet wurde.

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Bestimmungsgemäße Verwendung

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

Betriebsdauer, mittlere

Mean **t**ime **b**etween **f**ailures (MTBF) = mittlere Betriebsdauer zwischen Ausfällen.
Ist der Erwartungswert der Betriebsdauer

© ifm electronic gmbh - www.ifm.com

zwischen zwei aufeinanderfolgenden Ausfällen von Einheiten, die instand gesetzt werden.

Für Einheiten, die NICHT instandgesetzt werden, ist der Erwartungswert (Mittelwert) der Verteilung von Lebensdauern die mittlere Lebensdauer →MTTF.

Betriebssystem

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und der Anwender-Software.

Bootloader

Im Auslieferungszustand enthalten ecomatmobile-Geräte nur den Bootloader. Der Bootloader ist ein Startprogramm, mit dem das Betriebssystem (= Laufzeitsystem) und das Applikations-Programm auf dem Gerät nachgeladen werden können.

Der Bootloader enthält Grundroutinen...
- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Betriebssystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Bus

Serielle Datenübertragung mehrerer Teilnehmer an derselben Leitung.

CAN

CAN = **C**ontroller **A**rea **N**etwork

CAN gilt als Feldbussystem für größere Datenmengen, das prioritätengesteuert arbeitet. Gibt es in verschiedenen Varianten z.B. als "CANopen" oder "CAN in Automation (CiA)."

CAN-Stack

CAN-Stack = Stapel von CAN-Datenübertragungs-Aufträgen.

CCF

Common **c**ause **f**ailure = Ausfall in Folge von gemeinsamer Ursache
Ausfälle verschiedener Einheiten aufgrund eines gemeinsamen Ereignisses, wobei diese Ausfälle nicht auf gegenseitige Ursachen beruhen.

CiA

CiA = CAN in Automation e.V.

Anwender- und Herstellerorganisation in Deutschland / Erlangen. Definitions- und Kontrollorgan für CAN und CAN-basierende Netzwerkprotokolle.

Homepage → http://www.can-cia.org

CiA DS 304

DS = **D**raft **S**tandard

CAN-Geräteprofil CANopen-Safety für sicherheitsgerichtete Kommunikation.

CiA DS 401

DS = **D**raft **S**tandard

CAN-Geräteprofil für digitale und analoge E/A-Baugruppen

CiA DS 402

DS = **D**raft **S**tandard

CAN-Geräteprofil für Antriebe

CiA DS 403

DS = **D**raft **S**tandard

CAN-Geräteprofil für Bediengeräte

CiA DS 404

DS = **D**raft **S**tandard

CAN-Geräteprofil für Messtechnik und Regler

CiA DS 405

DS = **D**raft **S**tandard

Spezifikation zur Schnittstelle zu programmierbaren Steuerungen (IEC 61131-3)

CiA DS 406

DS = **D**raft **S**tandard

CAN-Geräteprofil für Drehgeber / Encoder

© ifm electronic gmbh | www.ifm.com

CiA DS 407

DS = **D**raft **S**tandard

CAN-Applikations-Profil für den öffentlichen Nahverkehr

COB-ID

COB = **C**ommunication **Ob**ject = Kommunikationsobjekt
ID = **Id**entifier = Kennung

Über den COB-ID unterscheiden die Teilnehmer die verschiedenen auszutauschenden Nachrichten.

CoDeSys

CoDeSys ist eingetragene Marke der 3S – Smart Software Solutions GmbH, Deutschland

"CoDeSys for Automation Alliance" vereinigt Firmen der Automatisierungsindustrie, deren Hardwaregeräte alle mit dem weit verbreiteten IEC 61131-3 Entwicklungswerkzeug CoDeSys programmiert werden.

Homepage → http://www.3s-software.com

CRC

CRC = **C**yclic **R**edundancy **C**heck = zyklische Redundanzprüfung

CRC ist ein Verfahren aus der Informationstechnik zur Bestimmung eines Prüfwerts für Daten, um Fehler bei der Übertragung oder Duplizierung von Daten erkennen zu können.

Vor Beginn der Übertragung eines Blocks der Daten wird ein CRC-Wert berechnet. Nach Abschluss der Transaktion wird am Zielort der CRC-Wert erneut berechnet. Anschließend werden diese beiden Prüfwerte verglichen.

CSV-Datei

CSV = **C**omma **S**eparated **V**alues (auch: **C**haracter **S**eparated **V**alues)
Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.
• Die Dateinamen-Erweiterung lautet `.csv`.
• CoDeSys erwartet als Trennzeichen zwischen den Spalten der Quell-Tabelle ein Semikolon (;).
• CoDeSys erzeugt als Endezeichen einer Zeichenkette (String) ein Null-Byte (NUL).
• Jeder Datensatz (jede zu übertragende Tabellenzeile) sollte die gleiche Anzahl von Tabellenspalten haben.

Beispiel: Quell-Tabelle mit Zahlenwerten:

Wert 1.0	Wert 1.1	Wert 1.2	Wert 1.3
Wert 2.0	Wert 2.1	Wert 2.2	Wert 2.3
Wert 3.0	Wert 3.1	Wert 3.2	Wert 3.3

Daraus entsteht folgende CSV-Datei:

```
Wert 1.0;Wert 1.1;Wert 1.2;Wert 1.3
Wert 2.0;Wert 2.1;Wert 2.2;Wert 2.3
Wert 3.0;Wert 3.1;Wert 3.2;Wert 3.3
```

Datentyp

Daten Typ	min. Wert	max. Wert	Größe im Speicher
BOOL	FALSE	TRUE	08 Bits = 1 Byte
BYTE	0	255	08 Bits = 1 Byte
WORD	0	65 535	16 Bits = 2 Bytes
DWORD	0	4 294 967 295	32 Bits = 4 Bytes
SINT	-128	127	08 Bits = 1 Byte
USINT	0	255	08 Bits = 1 Byte
INT	-32 768	32 767	16 Bits = 2 Bytes
UINT	0	65 535	16 Bits = 2 Bytes
DINT	-2 147 483 648	2 147 483 647	32 Bits = 4 Bytes
UDINT	0	4 294 967 295	32 Bits = 4 Bytes
REAL	1,175494351e-38	3,402823466e+38	32 Bits = 4 Bytes
STRING			Anz. Zeichen + 1

DC

Direct **C**urrent = Gleichstrom

DC

Diagnostic **C**overage = Diagnose-Deckungsgrad
Der Diagnose-Deckungsgrad ist das Maß für die Wirksamkeit der Diagnose als Verhältnis der Ausfallrate der bemerkten gefahrbringenden Ausfälle und der Ausfallrate der gesamten gefahrbringenden Ausfälle:

Formel: DC = Ausfallrate bemerkte gefahrbringende Ausfälle / Ausfallrate gesamte gefahrbringende Ausfälle

© ifm electronic gmbh | www.ifm.com

Bezeichnung	Bereich
kein	DC < 60 %
niedrig	60 % ≤ DC < 90 %
mittel	90 % ≤ DC < 99 %
hoch	99 % ≤ DC

Tabelle: Diagnose-Deckungsgrad DC

Für die in der Tabelle gezeigten Grenzwerte wird eine Genauigkeit von 5 % angenommen.

Der Diagnose-Deckungsgrad kann für das gesamte sicherheitsgerichtete System ermittelt werden oder nur für Teile des sicherheitsgerichteten Systems.

Diagnose

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:
- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.

Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.

Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

Diagnose-Deckungsgrad

→ DC

Dither

to dither (engl.) = schwanken / zittern

Dither ist ein Bestandteil der PWM-Signale zum Ansteuern von Hydraulik-Ventilen. Für die elektromagnetischen Antriebe von Hydraulik-Ventilen hat sich herausgestellt, dass sich die Ventile viel besser regeln lassen, wenn das Steuersignal (PWM-Impulse) mit einer bestimmten Frequenz der PWM-Frequenz überlagert wird. Diese Dither-Frequenz muss ein ganzzahliger Teil der PWM-Frequenz sein.

diversitär

Unter Diversität (Vielfalt) versteht man in der Technik eine Strategie zur Erhöhung der Ausfallsicherheit.

Dabei werden Systeme redundant ausgelegt, allerdings werden bewusst verschiedene Realisierungen und keine baugleichen Einzelsysteme verwendet. Man geht davon aus, dass Systeme, die das Gleiche leisten, aber unterschiedlich realisiert sind, auch gegen unterschiedliche Störungen empfindlich oder unempfindlich sind und daher möglichst nicht alle gleichzeitig ausfallen.

Die konkrete Realisierung kann je nach Einsatzgebiet und geforderter Sicherheit unterschiedlich aussehen:

- Verwendung von Bauteilen verschiedener Hersteller,
- Nutzung unterschiedlicher Protokolle zur Steuerung von Geräten,
- Verwendung komplett unterschiedlicher Technologien, beispielsweise einer elektrischen und einer pneumatischen Steuerung,
- Verwendung unterschiedlicher Messmethoden (Strom, Spannung),
- zwei Kanäle mit gegenläufigen Werteverläufen:
  Kanal A: 0...100 %
  Kanal B: 100...0 %

DRAM

DRAM = **D**ynamic **R**andom **A**ccess **M**emory

Technologie für einen elektronischen Speicherbaustein mit wahlfreiem Zugriff (Random Access Memory, RAM). Das speichernde Element ist dabei ein Kondensator, der entweder geladen oder entladen ist. Über einen Schalttransistor wird er zugänglich und entweder ausgelesen oder mit neuem Inhalt beschrieben. Der Speicherinhalt ist flüchtig: die gespeicherte Information geht bei fehlender Betriebsspannung oder zu später Wiederauffrischung verloren.

© ifm electronic gmbh | www.ifm.com

DTC

DTC = **D**iagnostic **T**rouble **C**ode = Fehler-Code
Störungen und Fehler werden über zugeordnete Nummern – den DTCs – verwaltet und gemeldet.

ECU

(1) **E**lectronic **C**ontrol **U**nit = Steuergerät oder Mikrocontroller
(2) **E**ngine **C**ontrol **U**nit = Steuergerät eines Motors

EDS-Datei

EDS = **E**lectronic **D**ata **S**heet = elektronisch hinterlegtes Datenblatt, z.B. für:

- Datei für das Objektverzeichnis im Master
- CANopen-Gerätebeschreibungen

Via EDS können vereinfacht Geräte und Programme ihre Spezifikationen austauschen und gegenseitig berücksichtigen.

Embedded Software

System-Software, Grundprogramm im Gerät, praktisch das Betriebssystem.

Die Firmware stellt die Verbindung her zwischen der Hardware des Gerätes und der Anwender-Software. Diese Software wird vom Hersteller der Steuerung als Teil des Systems geliefert und kann vom Anwender nicht verändert werden.

EMCY

Abkürzung für Emergency (engl.) = Notfall

EMV

EMV = **E**lektro-**M**agnetische **V**erträglichkeit

Gemäß der EG-Richtlinie (2004/108/EG) zur elektromagnetischen Verträglichkeit (kurz EMV-Richtlinie) werden Anforderungen an die Fähigkeit von elektrischen und elektronischen Apparaten, Anlagen, Systemen oder Bauteilen gestellt, in der vorhandenen elektromagnetischen Umwelt zufriedenstellend zu arbeiten. Die Geräte dürfen ihre Umgebung nicht stören und dürfen sich von äußerlichen elektromagnetischen Störungen nicht ungünstig beeinflussen lassen.

Erstfehler-Eintrittszeit

Das ist die Zeit bis zum ersten Versagen eines Sicherheitselements.

Im Zeitraum von maximal 30 s überprüft das Betriebssystem mittels interner Überwachungs- und Testroutinen die Steuerung.

Diese „Testzykluszeit“ muss kleiner sein als die statistische Erstfehler-Eintrittszeit für die Applikation.

Ethernet

Das Ethernet ist eine weit verbreitete, herstellerneutrale Technologie, mit der im Netzwerk Daten mit einer Geschwindigkeit von 10 oder 100 Millionen Bit pro Sekunde (Mbps) übertragen werden können. Das Ethernet gehört zu der Familie der sogenannten „bestmöglichen Datenübermittlung“ auf einem nicht exklusiven Übertragungsmedium. 1972 entwickelt, wurde das Konzept 1985 als IEEE 802.3 spezifiziert.

EUC

EUC = „equipment under control" (kontrollierte Einrichtung)

EUC ist eine Einrichtung, Maschine, Gerät oder Anlage, verwendet zur Fertigung, Stoffumformung, zum Transport, zu medizinischen oder anderen Tätigkeiten (→ IEC 61508-4, Abschnitt 3.2.3). Das EUC umfasst also alle Einrichtungen, Maschinen, Geräte oder Anlagen, die Gefährdungen verursachen können und für die sicherheitsgerichtete Systeme erforderlich sind.

Falls eine vernünftigerweise vorhersehbare Aktivität oder Inaktivität zu durch das EUC verursachten Gefährdungen mit unvertretbarem Risiko führt, sind Sicherheitsfunktionen erforderlich, um einen sicheren Zustand für das EUC zu erreichen oder aufrecht zu erhalten. Diese Sicherheitsfunktionen werden durch ein oder mehrere sicherheitsgerichtete Systeme ausgeführt.

Fehlanwendung

Das ist die Verwendung eines Produkts in einer Weise, die vom Konstrukteur nicht vorgesehen ist. Eine Fehlanwendung führt

© ifm electronic gmbh | www.ifm.com

meist zu einer Gefährdung von Personen oder Sachen.

Vor vernünftigerweise, vorhersehbaren Fehlanwendungen muss der Hersteller des Produkts in seinen Benutzerinformationen warnen.

Fehler

Ein Fehler ist die Unfähigkeit einer Einheit, eine geforderte Funktion auszuführen.

Kein Fehler ist diese Unfähigkeit während vorbeugender Wartung oder anderer geplanter Handlungen oder aufgrund des Fehlers externer Mittel.

Ein Fehler ist oft das Resultat eines Ausfalls der Einheit selbst, kann aber ohne vorherigen Ausfall bestehen.

In der ISO 13849-1 ist mit "Fehler" der "zufällige Fehler" gemeint.

Fehler-Toleranzzeit

Das ist die maximale Zeit, die zwischen dem Entstehen eines Fehlers und der Einnahme des sicheren Zustandes in der Applikation vergehen darf, ohne dass eine Gefahr für Personen zu befürchten ist.

Dabei ist die maximale Zykluszeit des Applikations-Programms (im ungünstigsten Fall 100 ms, → ***Verhalten des Watchdog***) und die möglichen Verzögerungs- und Reaktionszeiten durch Abschaltglieder zu berücksichtigen.

Die sich daraus ergebende Gesamtzeit muss kleiner sein als die Fehler-Toleranzzeit der Applikation.

FiFo

FiFo (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Firmware

System-Software, Grundprogramm im Gerät, praktisch das Betriebssystem.

Die Firmware stellt die Verbindung her zwischen der Hardware des Gerätes und der Anwender-Software. Diese Software wird vom Hersteller der Steuerung als Teil des Systems geliefert und kann vom Anwender nicht verändert werden.

Flash-Speicher

Flash-ROM (oder Flash-EPROM oder Flash-Memory) kombiniert die Vorteile von Halbleiterspeicher und Festplatten. Wie jeder andere Halbleiterspeicher kommt Flash-Speicher ohne bewegliche Teile aus. Und die Daten bleiben wie bei einer Festplatte auch nach dem Ausschalten erhalten.

Der Flash-ROM hat sich aus dem EEPROM (**E**lectrical **E**rasable and **P**rogrammable **R**ead-**O**nly **M**emory) entwickelt. Beim Flash-ROM ist die Speicherung von Daten funktionell identisch wie beim EEPROM. Die Daten werden allerdings wie bei einer Festplatte blockweise in Datenblöcken zu 64, 128, 256, 1024, ... Byte zugleich geschrieben und gelöscht.

Vorteile von Flash-Speicher

- Die gespeicherten Daten bleiben auch bei fehlender Versorgungsspannung erhalten.
- Wegen fehlender beweglicher Teile ist Flash geräuschlos, unempfindlich gegen Erschütterungen und magnetische Felder.
- Im Vergleich zu Festplatten haben Flash-Speicher eine sehr kurze Zugriffszeit. Lese- und Schreibgeschwindigkeit sind über den gesamten Speicherbereich weitestgehend konstant.
- Die erreichbare Speichergröße ist durch die einfache und platzsparende Anordnung der Speicherzellen nach oben offen.

Nachteile von Flash-Speicher

- Begrenzte Zahl von Schreib- bzw. Löschvorgängen, die eine Speicherzelle vertragen kann:
  - Multi-Level-Cells: typ. 10 000 Zyklen
  - Single-Level-Cells: typ. 100 000 Zyklen
- Da ein Schreibvorgang Speicherblöcke zwischen 16 und 128 kByte gleichzeitig beschreibt, werden auch Speicherzellen beansprucht, die gar keiner Veränderung bedürfen.

FMEA

FMEA = **F**ailure **M**ode and **E**ffects **A**nalysis = **F**ehler-**M**öglichkeits- und **E**influss-**A**nalyse

Methode der Zuverlässigkeitstechnik, um potenzielle Schwachstellen zu finden. Im

© ifm electronic gmbh | www.ifm.com

Rahmen des Qualitäts- oder Sicherheitsmanagements wird die FMEA zur Fehlervermeidung und Erhöhung der technischen Zuverlässigkeit vorbeugend eingesetzt.

FRAM

FRAM, oder auch FeRAM, bedeutet **Fe**rroelectric **R**andom **A**ccess **M**emory. Der Speicher- und Löschvorgang erfolgt durch eine Polarisationsänderung in einer ferroelektrischen Schicht.

Vorteile von FRAM gegenüber herkömmlichen Festwertspeichern:

- nicht flüchtig,
- kompatibel zu gängigen EEPROMs, jedoch:
- Zugriffszeit ca. 100 ns,
- fast unbegrenzt viele Zugriffszyklen möglich.

Funktionale Sicherheit

Teil der Gesamtsicherheit, bezogen auf das →EUC und das EUC-Leit- oder Steuerungssystem, die von der korrekten Funktion des elektrischen oder elektronischen sicherheitsgerichteten Systems, sicherheitsgerichteten Systemen anderer Technologien und externer Einrichtungen zur Risikominderung abhängt.

Gebrauchsdauer Tm

Die Gebrauchsdauer T_M ist der Zeitraum, der die vorgegebene Verwendung der SRP/CS abdeckt.

Gefährdung

Mit Gefährdung bezeichnet man eine potentielle Schadensquelle.

Man unterscheidet den Ursprung der Gefährdung, z.B.:
- mechanische Gefährdung,
- elektrische Gefährdung,

oder die Art des zu erwartenden Schadens, z.B.:
- Gefährdung durch elektrischen Schlag,
- Gefährdung durch Schneiden,
- Gefährdung durch Vergiftung.

Die Gefährdung im Sinne dieser Definition ist bei der bestimmungsgemäßen Verwendung der Maschine entweder dauerhaft vorhanden, z.B.:
- Bewegung von gefährdenden beweglichen Teilen,
- Lichtbogen beim Schweißen,
- ungesunde Körperhaltung,
- Geräusch-Emission,
- hohe Temperatur,

oder die Gefährdung kann unerwartet auftreten, z.B.:
- Explosion,
- Gefährdung durch Quetschen als Folge eines unbeabsichtigten / unerwarteten Anlaufs,
- Herausschleudern als Folge eines Bruchs,
- Stürzen als Folge von Geschwindigkeitsänderung.

Heartbeat

Heartbeat (engl.) = Herzschlag

Die Teilnehmer senden regelmäßig kurze Signale. So können die anderen Teilnehmer prüfen, ob ein Teilnehmer ausgefallen ist. Dazu ist kein Master erforderlich.

HMI

HMI = **H**uman **M**achine **I**nterface = Mensch-Maschine-Schnittstelle

ID - Identifier

ID = **Id**entifier = Kennung

Name zur Unterscheidung der an einem System angeschlossenen Geräte / Teilnehmer oder der zwischen den Teilnehmern ausgetauschten Nachrichtenpakete.

IEC-User-Zyklus

IEC-User-Zyklus = SPS-Zyklus im CoDeSys-Applikations-Programm.

IP-Adresse

IP = **I**nternet **P**rotocol = Internet-Protokoll

Die IP-Adresse ist eine Nummer, die zur eindeutigen Identifizierung eines Internet-Teilnehmers notwendig ist. Zur besseren Übersicht wird die Nummer in 4 dezimalen Werten geschrieben, z. B. 127.215.205.156.

© ifm electronic gmbh / www.ifm.com

ISO 11898

Norm: "Straßenfahrzeuge – CAN-Protokoll"

Teil 1: "Bit-Übertragungsschicht und physikalische Zeichenabgabe"

Teil 2: "High-speed medium access unit"

Teil 3: "Fehlertolerante Schnittstelle für niedrige Geschwindigkeiten"

Teil 4: "Zeitgesteuerte Kommunikation"

Teil 5: "High-speed medium access unit with low-power mode"

ISO 11992

Norm: "Straßenfahrzeuge – Austausch von digitalen Informationen über elektrische Verbindungen zwischen Zugfahrzeugen und Anhängefahrzeugen"

Teil 1: "Bit-Übertragungsschicht und Sicherungsschicht"

Teil 2: "Anwendungsschicht für die Bremsausrüstung"

Teil 3: "Anwendungsschicht für andere als die Bremsausrüstung"

Teil 4: "Diagnose"

ISO 16845

Norm: "Straßenfahrzeuge – Steuergerätenetz (CAN) – Prüfplan zu Konformität"

Kategorie (CAT)

Einstufung der sicherheitsrelevante Teile einer Steuerung bezüglich ihres Widerstandes gegen Fehler und ihres nachfolgenden Verhaltens bei einem Fehler. Diese Sicherheit wird erreicht durch die Struktur der Anordnung der Teile, die Fehlererkennung und/oder ihre Zuverlässigkeit. (→ EN 954)

Klemme 15

Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

Lebensdauer, mittlere

Mean **t**ime **t**o **d**angerous **f**ailure = erwartete mittlere Dauer bis zum gefahrbringenden Ausfall.

Bezeichnung	Bereich
niedrig	3 Jahre ≤ $MTTF_d$ < 10 Jahre
mittel	10 Jahre ≤ $MTTF_d$ < 30 Jahre
hoch	30 Jahre ≤ $MTTF_d$ ≤ 100 Jahre

Tabelle: Mittlere Zeit jedes Kanals bis zum gefahrbringenden Ausfall $MTTF_d$

LED

LED = **L**ight **E**mitting **D**iode = Licht aussendende Diode

Leuchtdiode, auch Luminiszenzdiode, ein elektronisches Element mit hoher, farbiger Leuchtkraft auf kleinem Volumen bei vernachlässigbarer Verlustleistung.

Link

Ein Link ist ein Querverweis zu einer anderen Stelle im Dokument oder auf ein externes Dokument.

LSB

Least **S**ignificant **B**it/Byte = Niederwertigstes Bit/Byte in einer Reihe von Bit/Bytes.

MAC-ID

MAC = **M**anufacturer‘s **A**ddress **C**ode = Hersteller-Seriennummer

→ID = **Id**entifier = Kennung

Jede Netzwerkkarte verfügt über eine so genannte MAC-Adresse, ein unverwechselbarer, auf der ganzen Welt einzigartiger Zahlencode – quasi eine Art Seriennummer. So eine MAC-Adresse ist eine Aneinanderreihung von 6 Hexadezimalzahlen, etwa "00-0C-6E-D0-02-3F".

Master

Wickelt die komplette Organisation auf dem Bus ab. Der Master entscheidet über den zeitlichen Buszugriff und fragt die →Slaves zyklisch ab.

© ifm electronic gmbh | www.ifm.com

MMI

MMI = **M**ensch-**M**aschine-**I**nterface
→ *HMI* (→ Seite 307)

MRAM

MRAM bedeutet **M**agnetoresistive **R**andom **A**ccess **M**emory. Die Informationen werden mit magnetischen Ladungselementen gespeichert. Dabei wird die Eigenschaft bestimmter Materialien ausgenutzt, die ihren elektrischen Widerstand unter dem Einfluss magnetischer Felder ändern.

Vorteile von MRAM gegenüber herkömmlichen Festwertspeichern:

- nicht flüchtig (wie FRAM), jedoch:
- Zugriffszeit nur ca. 35 ns,
- unbegrenzt viele Zugriffszyklen möglich.

MSB

Most **S**ignificant **B**it/Byte = Höchstwertiges Bit/Byte einer Reihe von Bits/Bytes.

MTBF

Mean **t**ime **b**etween **f**ailures (MTBF) = mittlere Betriebsdauer zwischen Ausfällen.
Ist der Erwartungswert der Betriebsdauer zwischen zwei aufeinanderfolgenden Ausfällen von Einheiten, die instand gesetzt werden.

Für Einheiten, die NICHT instandgesetzt werden, ist der Erwartungswert (Mittelwert) der Verteilung von Lebensdauern die mittlere Lebensdauer →MTTF.

MTTF

Mean **t**ime **to** **f**ailure (MTTF) = mittlere Dauer bis zum Ausfall oder: mittlere Lebensdauer.

MTTFd

Mean **t**ime **to** **d**angerous **f**ailure = erwartete mittlere Dauer bis zum gefahrbringenden Ausfall.

Bezeichnung	Bereich
niedrig	3 Jahre ≤ $MTTF_d$ < 10 Jahre
mittel	10 Jahre ≤ $MTTF_d$ < 30 Jahre
hoch	30 Jahre ≤ $MTTF_d$ ≤ 100 Jahre

Tabelle: Mittlere Zeit jedes Kanals bis zum gefahrbringenden Ausfall $MTTF_d$

Muting

Mit Muting bezeichnet man die vorübergehende und automatische Unterdrückung einer Sicherheitsfunktion durch das SRP/CS.

Beispiel: Der Sicherheits-Lichtvorhang ist überbrückt, wenn die schließenden Werkzeuge unter einen fingersicheren Abstand zueinander gelangt sind. Die bedienende Person kann nun gefahrlos an die Maschine herantreten und das Werkstück führen.

NMT

NMT = **N**etwork **M**anagemen**t** = Netzwerk-Verwaltung (hier: im CAN-Bus)

Der NMT-Master steuert die Betriebszustände der NMT-Slaves.

Node

Node (engl.) = Knoten. Damit ist ein Teilnehmer im Netzwerk gemeint.

Node Guarding

Node (engl.) = Knoten, hier: Netzwerkteilnehmer
Guarding (engl.) = Schutz

Parametrierbare, zyklische Überwachung von jedem entsprechend konfigurierten Slave. Der Master prüft, ob die Slaves rechtzeitig antworten. Die Slaves prüfen, ob der Master regelmäßig anfragt. Somit können ausgefallene Netzwerkteilnehmer schnell erkannt und gemeldet werden.

Obj / Objekt

Oberbegriff für austauschbare Daten / Botschaften innerhalb des CANopen-Netzwerks.

© ifm electronic gmbh | www.ifm.com

Objektverzeichnis

Das **Ob**jekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OBV

Das **Ob**jekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

operational

Operational (engl.) = betriebsbereit

Betriebszustand eines CANopen-Teilnehmers. In diesem Modus können SDOs, NMT-Kommandos und PDOs übertragen werden.

PC-Karte

→ PCMCIA-Karte

PCMCIA-Karte

PCMCIA = Personal Computer Memory Card International Association, ein Standard für Erweiterungskarten mobiler Computer.
Seit der Einführung des Cardbus-Standards 1995 werden PCMCIA-Karten auch als PC-Karte (engl.: PC Card) bezeichnet.

PDM

PDM = **P**rocess and **D**ialog **M**odule = **P**rozess- und **D**ialog-**M**onitor

Gerät zur Kommunikation des Bedieners mit der Maschine / Anlage.

PDO

PDO = **P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten.

Die zeitkritischen Prozessdaten werden mit Hilfe der "Process Data Objects" (PDOs) übertragen. Die PDOs können beliebig zwischen den einzelnen Knoten ausgetauscht werden (PDO-Linking). Zusätzlich wird festgelegt, ob der Datenaustausch ereignisgesteuert (asynchron) oder synchronisiert erfolgen soll. Je nach der Art der zu übertragenden Daten kann die richtige Wahl der Übertragungsart zu einer erheblichen Entlastung des CAN-Bus führen.

Diese Dienste sind vom Protokoll her nicht bestätigte Dienste, d.h. es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

PDU

PDU = **P**rotocol **D**ata **U**nit = Protokoll-Daten-Einheit

Die PDU ist ein Begriff aus dem CAN-Protokoll SAE J1939. Sie bezeichnet einen Bestandteil der Ziel- oder Quelladresse.

Performance-Level

Performance-**L**evel
Ist nach ISO 13849-1 eine Einstufung (PL a...e) der Fähigkeit von sicherheitsrelevanten Teilen einer Steuerung, eine Sicherheitsfunktion unter vorhersehbaren Bedingungen auszuführen.

PES

Programable **e**lectronic **s**ystem = Programmierbares elektronisches System
Ein programmierbares elektronisches System ist ein System ...
- zur Steuerung, zum Schutz oder zur Überwachung,
- auf der Basis einer oder mehrerer programmierbarer Geräte,
- einschließlich aller Elemente dieses Systems, wie Ein- und Ausgabegeräte.

PGN

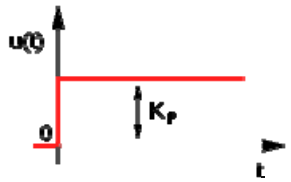
PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer
PGN = PDU Format (PF) + PDU Source (PS)

Die Parameter-Gruppennummer ist ein Begriff aus dem CAN-Protokoll SAE J1939. Sie fasst die Teiladressen PF und PS zusammen.

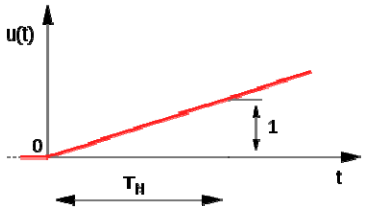
© ifm electronic gmbh | www.ifm.com

PID-Regler

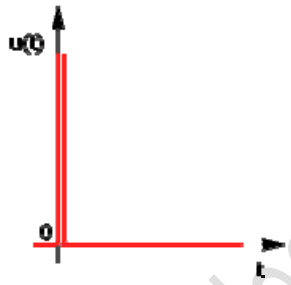
P = Proportional-Anteil
Der P-Regler besteht ausschließlich aus einem proportionalen Anteil der Verstärkung K_p. Mit seinem Ausgangssignal ist er proportional dem Eingangssignal.



I = Integral-Anteil
Ein I-Regler wirkt durch zeitliche Integration der Regelabweichung auf die Stellgröße mit der Gewichtung durch die Nachstellzeit T_N



D = Differential-Anteil
Der D-Regler reagiert nicht auf die Regelabweichung, sondern nur auf deren Änderungsgeschwindigkeit.



Piktogramm

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln.

→ Kapitel *Was bedeuten die Symbole und Formatierungen?* (→ Seite 5)

PL

Performance-**L**evel
Ist nach ISO 13849-1 eine Einstufung (PL a...e) der Fähigkeit von sicherheitsrelevanten Teilen einer Steuerung, eine Sicherheitsfunktion unter vorhersehbaren Bedingungen auszuführen.

PLr

Mit dem "erforderlichen Performance-Level" PL_r wird nach ISO 13849 die erforderliche Risikominderung für jede Sicherheitsfunktion erreicht.

Für jede gewählte Sicherheitsfunktion, die durch ein SRP/CS ausgeführt wird, muss ein PL_r festgelegt und dokumentiert werden. Die Bestimmung des PL_r ist das Ergebnis der Risikobeurteilung, bezogen auf den Anteil der Risikominderung durch die sicherheitsrelevanten Teile der Steuerung.

Pre-Op

Pre-Op = PRE-OPERATIONAL mode (engl.) = Zustand vor betriebsbereit

Betriebszustand eines CANopen-Teilnehmers. Nach dem Einschalten der Versorgungsspannung geht jeder Teilnehmer automatisch in diesem Zustand. Im CANopen-Netz können in diesem Modus nur SDOs und NMT-Kommandos übertragen werden, jedoch keine Prozessdaten.

prepared

prepared (engl.) = vorbereitet (auch: angehalten)

Betriebszustand eines CANopen-Teilnehmers. In diesem Modus werden nur NMT-Kommandos übertragen.

Programmiersprache, sicherheitsrelevant

Für sicherheitsrelevante Applikationen sollten nur folgende Programmiersprachen verwendet werden:

- Programmiersprache mit eingeschränktem Sprachumfang (LVL = **l**imited **v**ariability **l**anguage), kann vordefinierte, applikations-spezifische Bibliotheksfunktionen kombinieren.
  In CoDeSys sind das Kontaktplan KOP (Ladder Diagram LD) und Funktionsplan FUP (Function block diagram FBD).
- Programmiersprache mit nicht eingeschränktem Sprachumfang (FVL = **f**ull **v**ariability **l**anguage), kann einen großen Bereich von Funktionen kombinieren.

© ifm electronic gmbh - www.ifm.com

Dazu gehören z.B. C, C++, Assembler. In CoDeSys ist das Strukturierter Text (ST).

► Strukturierter Text ist ausschließlich in gesonderten, zertifizierten Funktionen zu empfehlen, normalerweise in Embedded Software.

► Im "normalen" Applikations-Programm sollten nur KOP (LD) und FUP (FBD) eingesetzt werden. Damit sollen die folgenden Mindestanforderungen erfüllt werden können.

Generell werden folgende Mindestanforderungen an sicherheitsrelevante Applikations-Software (SRASW) gestellt:

► Programm modular und klar strukturieren. Folge: einfache Testbarkeit.

► Funktionen verständlich darstellen:
- für den Operator auf dem Bildschirm (Navigation),
- Lesbarkeit des späteren Dokumentationsausdrucks.

► Symbolische Variablen verwenden (keine IEC-Adressen).

► Variablennamen und Kommentare aussagekräftig formulieren.

► Einfache Funktionen verwenden (keine indirekte Adressierung, keine Variablenfelder).

► Defensiv programmieren.

► Leichtes Erweitern oder Anpassen des Programms ermöglichen.

Prozessabbild

Mit Prozessabbild bezeichnet man den Zustand der Ein- und Ausgänge, mit denen die SPS innerhalb eines Zyklusses arbeitet.

- Am Zyklus-Beginn liest die SPS die Zustände aller Eingänge in das Prozessabbild ein.
  Während des Zyklusses kann die SPS Änderungen an den Eingängen nicht erkennen.
- Im Laufe des Zyklusses werden die Ausgänge nur virtuell (im Prozessabbild) geändert.
- Am Zyklus-Ende schreibt die SPS die virtuellen Ausgangszustände auf die realen Ausgänge.

PWM

PWM = Puls-Weiten-Modulation

Via PWM kann ein (vom Gerät dazu befähigter) digitaler Ausgang mittels regelmäßiger, schneller Impulse eine beinahe analoge Spannung ausgeben. Bei dem PWM-Ausgangssignal handelt es sich um ein getaktetes Signal zwischen GND und Versorgungsspannung.

Innerhalb einer festen Periode (PWM-Frequenz) wird das Puls-/Pausenverhältnis variiert. Durch die angeschlossene Last stellt sich je nach Puls-/Pausenverhältnis der entsprechende Effektivstrom ein.
→ Kapitel *PWM-Signalverarbeitung* (→ Seite 228)

Ratio

Ratio (lat.) = Verhältnis

Messungen können auch ratiometrisch erfolgen = Verhältnismessung. Das Eingangssignal erzeugt ein Ausgangssignal, das in einem bestimmten Verhältnis zu ihm liegt. Das bedeutet, ohne zusätzliche Referenzspannung können analoge Eingangssignale ausgewertet werden. Ein Schwanken der Versorgungsspannung hat auf diesen Messwert dann keinen Einfluss.
→ Kapitel *Zählerfunktionen zur Frequenz- und Periodendauermessung* (→ Seite 215)

RAW-CAN

RAW-CAN bezeichnet das reine CAN-Protokoll, das ohne ein zusätzliches Kommunikationsprotokoll auf dem CAN-Bus (auf ISO/OSI-Schicht 2) arbeitet. Das CAN-Protokoll ist international nach ISO 11898-1 definiert und garantiert zusätzlich in ISO 16845 die Austauschbarkeit von CAN-Chips.

redundant

Redundanz ist das Vorhandensein von mehr als den notwendigen Mitteln, damit eine Funktionseinheit eine geforderte Funktion ausführt oder damit Daten eine Information darstellen können.

Man unterscheidet verschiedene Arten der Redundanz:

© ifm electronic gmbh | www.ifm.com

- Die funktionelle Redundanz zielt darauf ab, sicherheitstechnische Systeme mehrfach parallel auszulegen, damit beim Ausfall einer Komponente die anderen den Dienst gewährleisten.
- Zusätzlich versucht man, die redundanten Systeme voneinander räumlich zu trennen. Dadurch minimiert man das Risiko, dass sie einer gemeinsamen Störung unterliegen.
- Schließlich verwendet man manchmal Bauteile unterschiedlicher Hersteller, um zu vermeiden, dass ein systematischer Fehler sämtliche redundanten Systeme ausfallen lässt (diversitäre Redundanz).

Die Software von redundanten Systemen sollte sich möglichst in den folgenden Aspekten unterscheiden:

- Spezifikation (verschiedene Teams),
- Spezifikationssprache,
- Programmierung (verschiedene Teams),
- Programmiersprache,
- Compiler.

remanent

Remanente Daten sind gegen Datenverlust bei Spannungsausfall geschützt.

Z.B. kopiert das Betriebssystem die remanenten Daten automatisch in einen Flash-Speicher, sobald die Spannungsversorgung unter einen kritischen Wert sinkt. Bei Wiederkehr der Spannungsversorgung lädt das Betriebssystem die remanenten Daten zurück in den Arbeitsspeicher.

Dagegen sind die Daten im Arbeitsspeicher einer Steuerung flüchtig und bei Unterbrechung der Spannungsversorgung normalerweise verloren.

Restrisiko

Das ist das verbleibende Risiko, nachdem Schutzmaßnahmen ergriffen wurden. Vor dem Restrisiko muss in Betriebsanleitungen und an der Maschine deutlich gewarnt werden.

Risiko

Als Risiko gilt die Kombination der Wahrscheinlichkeit des Eintritts eines Schadens und des Ausmaßes des Schadens.

Risikoanalyse

Kombination aus ...

- Festlegung der Grenzen der Maschine (Verwendungszweck, zeitliche Grenzen),
- Identifizierung der Gefährdung (Eingreifen von Personen, Betriebszustände der Maschine, vorhersehbarer Missbrauch) und
- der Risikoeinschätzung (Verletzungsgrad, Schadensumfang, Häufigkeit und Dauer der Gefahr, Eintrittswahrscheinlichkeit, Möglichkeit zur Vermeidung oder Begrenzung des Schadens).

Risikobeurteilung

Das ist die Gesamtheit des Verfahrens, das die Risikoanalyse und die Risikobewertung umfasst.

Nach Maschinenrichtlinie 2006/42/EG gilt: "Der Hersteller einer Maschine oder sein Bevollmächtigter hat dafür zu sorgen, dass eine Risikobeurteilung vorgenommen wird, um die für die Maschine geltenden Sicherheits- und Gesundheitsanforderungen zu ermitteln. Die Maschine muss dann unter Berücksichtigung der Ergebnisse der Risikobeurteilung konstruiert und gebaut werden." (→ Anhang 1, Allgemeine Grundsätze)

Risikobewertung

Das ist die auf der Risikoanalyse beruhende Beurteilung, ob die Ziele zur Risikominderung erreicht wurden.

ro

ro = read only (engl.) = nur lesen

Unidirektionale Datenübertragung: Daten können nur gelesen werden, jedoch nicht verändert.

RTC

RTC = **R**eal **T**ime **C**lock = Echtzeituhr

Liefert (batteriegepuffert) aktuell Datum und Uhrzeit. Häufiger Einsatz beim Speichern von Fehlermeldungsprotokollen.

© ifm electronic gmbh / www.ifm.com

Rückstellung, manuell

Die manuelle Rückstellung ist eine interne Funktion des SRP/CS zum anuellen Wiederherstellen einer oder mehrerer Sicherheitsfunktionen. Wird vor dem Neustart einer Maschine verwendet.

rw

rw = read/write (engl.) = lesen und schreiben

Bidirektionale Datenübertragung: Daten können sowohl gelesen als auch verändert werden.

SAE J1939

Das Netzwerkprotokoll SAE J1939 beschreibt die Kommunikation auf einem CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B.Motordrehzahl, Temperatur) und Steuerungsinformationen.
→ CiA DS 402

Norm: "Recommended Practice for a Serial Control and Communications Vehicle Network"

Teil 2: "Agricultural and Forestry Off-Road Machinery Control and Communication Network"

Teil 3: "On Board Diagnostics Implementation Guide"

Teil 5: "Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide"

Teil 11: "Physical Layer – 250 kBits/s, Shielded Twisted Pair"

Teil 13: "Off-Board Diagnostic Connector"

Teil 15: "Reduced Physical Layer, 250 kBits/s, Un-Shielded Twisted Pair (UTP)"

Teil 21: "Data Link Layer"

Teil 31: "Network Layer"

Teil 71: "Vehicle Application Layer"

Teil 73: "Application Layer – Diagnostics"

Teil 81: "Network Management Protocol"

Schaden

Als Schaden bezeichnet man eine physische Verletzung oder Schädigung der Gesundheit.

Schutzmaßnahme

Maßnahme zur vorgesehenen Minderung des Risikos, z.B.:
- fehlerausschließender Entwurf,
- technische Schutzmaßnahme (trennende Schutzeinrichtung),
- ergänzende Schutzmaßnahme (Benutzerinformation),
- persönliche Schutzausrüstung (Helm, Schutzbrille).

SCT

Bei CANopen-Safety überprüft die Sicherheits-Zykluszeit SCT (**S**afeguard **c**ycle **t**ime) die korrekte Funktion der periodischen Übertragung (Daten-Refresh) der SRDOs. Die Daten müssen innerhalb der eingestellten Zeit wiederholt worden sein, um gültig zu sein. Andernfalls signalisiert die empfangene Steuerung einen Fehler und geht in den sicheren Zustand (= Ausgänge abgeschaltet).

SD-Card

Eine SD Memory Card (Kurzform für **S**ecure **D**igital Memory Card; deutsch Sichere digitale Speicherkarte) ist ein digitales Speichermedium, das nach dem Prinzip der Flash-Speicherung arbeitet.

SDO

SDO = **S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten.

SDO ist eine Spezifikation für eine herstellerunabhängige Datenstruktur zum einheitlichen Datenzugriff. Dabei fordern "Clients" die gewünschten Daten von "Servern" an. Die SDOs bestehen immer aus 8 Bytes. Längere Datenpakete werden auf mehrere Nachrichten verteilt.

Beispiele:

- Automatische Konfiguration aller Slaves über SDOs beim Systemstart.
- Auslesen der Fehlernachrichten aus dem Objektverzeichnis.

Jedes SDO wird auf Antwort überwacht und wiederholt, wenn sich innerhalb der Überwachungszeit der Slave nicht meldet.

© ifm electronic gmbh | www.ifm.com

Selbsttest

Testprogramm, das aktiv Komponenten oder Geräte testet. Das Programm wird durch den Anwender gestartet und dauert eine gewisse Zeit. Das Ergebnis davon ist ein Testprotokoll (Log-Datei), auf dem entnommen werden kann, was getestet wurde und ob das Ergebnis positiv oder negativ ist.

Sicherheitsfunktion

Der Ausfall einer Sicherheitsfunktion einer Maschine kann zum unmittelbar erhöhten Risiko führen. Der Konstrukteur einer solchen Maschine muss daher:
- einen Ausfall der Sicherheitsfunktion sicher verhindern,
- einen Ausfall der Sicherheitsfunktion rechtzeitig sicher erkennen,
- Maschine bei einem Ausfall der Sicherheitsfunktion rechtzeitig in einen sicheren Zustand bringen.

Sicherheits-Normentypen

Sicherheitsnormen auf dem Gebiet der Maschinen sind wie folgt strukturiert:

Typ-A-Normen (Sicherheits-Grundnormen) behandeln Grundbegriffe, Entwurfsleitsätze und allgemeine Aspekte, die auf Maschinen angewendet werden können. Beispiele: Terminologie Methodik (ISO 12100-1), Technische Prinzipien (ISO 12100-2), Risikobeurteilung (ISO 14121), ...

Typ-B-Normen (Sicherheits-Fachgrundnormen) behandeln einen Sicherheitsaspekt oder eine Art von Schutzeinrichtungen, die für eine Reihe von Maschinen verwendet werden können.

- Typ-B1-Normen für bestimmte Sicherheitsaspekte. Beispiele: Sicherheits-Abstände (EN 294), Arm-/Hand-Geschwindigkeiten (EN 999), Sicherheitsbezogene Teile von Steuerungen (ISO 13849), Temperaturen, Lärm, ...
- Typ-B2-Normen für Schutzeinrichtungen. Beispiele: Not-Aus-Schaltungen ((ISO 13850), Zweihand-Schaltungen, trennende oder berührungslos wirkende Schutzeinrichtungen (ISO 61496), ...

Typ-C-Normen (Maschinensicherheitsnormen) behandeln detaillierte Sicherheitsanforderungen an eine bestimmte Maschine oder Maschinengruppen.

SIL

Der Sicherheits-Integritätslevel SIL ist nach IEC 62061 eine Einstufung (SIL CL 1...4) der Sicherheitsintegrität der Sicherheitsfunktionen. Er dient der Beurteilung elektrischer/elektronischer/programmierbar elektronischer (E/E/PE)-Systeme in Bezug auf die Zuverlässigkeit von Sicherheitsfunktionen. Aus dem angestrebten Level ergeben sich die sicherheitsgerichteten Konstruktionsprinzipien, die eingehalten werden müssen, damit das Risiko einer Fehlfunktion minimiert werden kann.

Slave

Passiver Teilnehmer am Bus, antwortet nur auf Anfrage des →Masters. Slaves haben im Bus eine eindeutige und einmalige →Adresse.

SRDO

Über SRDOs (**S**afety-**R**elated **D**ata **O**bjects = Sicherheitsrelevante Datenobjekte) werden die sicheren Daten ausgetauscht. Ein SRDO besteht immer aus zwei CAN-Nachrichten mit unterschiedlichen Identifiern:

- Nachricht 1 enthält die Originalanwenderdaten,
- Nachricht 2 enthält die gleichen Daten, die aber bitweise invertiert werden.

SRP/CS

Safety-**R**elated **P**art of a **C**ontrol **S**ystem = Sicherheitsrelevanter Teil einer Steuerung

SRP/CS ist ein Teil einer Steuerung, das auf sicherheitsgerichtete Eingangssignale reagiert und sicherheitsgerichtete Ausgangssignale erzeugt. Die Kombination sicherheitsrelevanter Teile einer Steuerung beginnt an dem Punkt, an dem sicherheitsgerichtete Signale erzeugt werden (einschließlich Betätiger z.B. eines Positionsschalters) und endet an den Ausgängen der Leistungssteuerungselemente (einschließlich z.B. der Hauptkontakte eines Schützes).

© ifm electronic gmbh | www.ifm.com

SRVT

Die sicherheitsrelevante Objekt-Gültigkeitsdauer SRVT (**S**afety-**R**elated **O**bject **V**alidation **T**ime) sorgt bei CANopen-Safety dafür, dass die Zeit zwischen den SRDO-Nachrichten-Paaren eingehalten wird:

Nur wenn die redundante, invertierte Nachricht innerhalb der eingestellten Zeit SRVT nach der Original-Nachricht übertragen wurde, sind die damit übertragenen Daten gültig. Andernfalls signalisiert die empfangende Steuerung einen Fehler und geht in den sicheren Zustand (= Ausgänge abgeschaltet).

Steuerungskonfiguration

Bestandteil der CoDeSys-Bedienoberfläche.

► Programmierer teilt dem Programmiersystem mit, welche Hardware programmiert werden soll.

> CoDeSys lädt die zugehörigen Bibliotheken.

> Lesen und schreiben der Peripherie-Zustände (Ein-/Ausgänge) ist möglich.

Symbole

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln.

→ Kapitel ***Was bedeuten die Symbole und Formatierungen?*** (→ Seite 5)

Systemvariable

Variable, auf die via IEC-Adresse oder Symbolname aus der SPS zugegriffen werden kann.

Target

Das Target gibt das Zielsystem an, auf dem das SPS-Programm laufen soll. Im Target sind die Dateien (Treiber und ggf. spezifische Hilfedateien) enthalten, die zum Programmieren und Parametrieren erforderlich sind.

TCP

Das **T**ransmission **C**ontrol **P**rotocol ist Teil der Protokollfamilie TCP/IP. Jede TCP/IP-Datenverbindung hat einen Sender und einen Empfänger. Dieses Prinzip ist eine verbindungsorientierte Datenübertragung. In der TCP/IP-Protokollfamilie übernimmt TCP als verbindungsorientiertes Protokoll die Aufgabe der Datensicherheit, der Datenflusssteuerung und ergreift Maßnahmen bei einem Datenverlust. (vgl.: →UDP)

Template

Template (englisch = Schablone)

Ist eine Vorlage, die mit Inhalten gefüllt werden kann.
Hier: Eine Struktur von vorkonfigurierten Software-Elementen als Basis für ein Applikations-Programm.

Testrate rt

Die Testrate r_t ist die Häufigkeit der automatischen Tests, um Fehler in einem SRP/CS rechtzeitig zu bemerken.

Überwachung

Die Überwachung ist eine Sicherheitsfunktion, die sicherstellt, dass eine Schutzmaßnahme eingeleitet wird, sobald Folgendes eintritt:

- Die Fähigkeit eines Bauteils oder eines Elements, seine Funktion auszuführen, wird vermindert.
- Die Betriebsbedingungen werden so verändert, dass das resultierende Risiko steigt.

UDP

UDP (**U**ser **D**atagram **P**rotocol) ist ein minimales, verbindungsloses Netzprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört. Aufgabe von UDP ist es, Daten, die über das Internet übertragen werden, der richtigen Applikation zukommen zu lassen.

Derzeit sind Netzwerkvariablen auf Basis von CAN und UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. In UDP sind diese als Broadcast-Telegramme realisiert, in CAN als

© ifm electronic gmbh | www.ifm.com

PDOs. Diese Dienste sind vom Protokoll her nicht bestätigte Dienste, d.h. es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

Verwendung, bestimmungsgemäß

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

Watchdog

Der Begriff Watchdog (englisch; Wachhund) wird verallgemeinert für eine Komponente eines Systems verwendet, die die Funktion anderer Komponenten beobachtet. Wird dabei eine mögliche Fehlfunktionen erkannt, so wird dies entweder signalisiert oder geeignete Programm-Verzweigungen eingeleitet. Das Signal oder die Verzweigungen dienen als Auslöser für andere kooperierende Systemkomponenten, die das Problem lösen sollen.

wo

wo = write only (engl.) = nur schreiben

Unidirektionale Datenübertragung: Daten können nur verändert werden, jedoch nicht gelesen.

Zustand, sicher

Der Zustand einer Maschine gilt als sicher, wenn von ihr keine Gefährdung mehr ausgeht. Dies ist meist der Fall, wenn alle gefahrbringenden Bewegungsmöglichkeiten abgeschaltet sind und nicht unerwartet wieder anlaufen können.

Zykluszeit

Das ist die Zeit für einen Zyklus. Das SPS-Programm läuft einmal komplett durch.

Je nach ereignisgesteuerten Verzweigungen im Programm kann dies unterschiedlich lange dauern.

© ifm electronic gmbh | www.ifm.com

15 Index

A

Abarbeitung der Programmbausteine 60
Abfrage des Slave-Gerätetyps 142
Abgrenzung zu anderen CANopen-Bibliotheken 134
Adressbelegung der Ausgänge (CR0401) 261
Adressbelegung der Ausgänge (CR0403) 261
Adressbelegung der Eingänge 260
Adressbelegung Ein-/Ausgänge 260
Adressbelegung und E/A-Betriebsarten 259
Adresse 301
Allgemeine Informationen 158
Allgemeine Übersicht 295
Allgemeines zu CAN 49
Allgemeines zu CANopen mit CoDeSys 132
Analogeingänge 25
Ändern der PDO-Eigenschaften zur Laufzeit 157
Anforderungsrate rd 301
Angaben zum Gerät 10
Angaben zur Software 14
Anhang 259
Anleitung 301
Applikations-Programm erstellen 45
Applikations-Software 301
Architektur 301
Aufbau des COB-ID 267
Aufbau einer EMCY-Nachricht 199
Aufbau einer Fehlernachricht 199
Aufbau von CANopen-Meldungen 266
Ausfall 301
Ausfall, gefahrbringend 301
Ausfall, systematischer 301
Ausgänge konfigurieren 29
Ausgangsfunktionen allgemein 223
Ausgangsgruppe 1 30
Ausgangsgruppe 2 31
Ausgangsgruppe 3 32
Automatische Datensicherung 237
Automatische Konfiguration von Slaves 143

B

Baud 301
Baustein-Ausgänge 62
Bausteine für J1939 (Empfangen) 121
Bausteine für J1939 (Senden) 111
Bausteine für J1939 (Verwaltung) 102
Baustein-Eingänge 61
Begrenzungen und Programmierhinweise 41
Beispiel
ausführliche Nachrichten-Dokumentation 99
kurze Nachrichten-Dokumentation 100
Variablenliste 155
Beispiel für ein Objektverzeichnis 153
Beispielablauf für Reaktion auf System-Fehler 48
Besonderheiten bei Netzwerkvariablen 163
Bestimmungsgemäße Verwendung 301
Betriebsdauer, mittlere 301
Betriebssystem 39, 302
Betriebssystem laden 39
Betriebszustände 35
Betriebszustände und Betriebssystem 35
Bibliotheken für CAN-Bausteine 63
Bootloader 302
Bootup-Nachricht 271
Bus 302
Busleitungslänge 55

C

CAN 302
CAN einsetzen 49
CAN für die Antriebstechnik 97
CAN Parameter
Alle SDOs erzeugen 138
Communication Cycle 139
DCF schreiben 138
Emergency Telegram 139
Knoten zurücksetzen 138
Nicht initialisieren 138
Nodeguarding- / Heartbeat-Einstellungen 138
Node-ID 138
Optionales Gerät 138
CAN_ENABLE 67
CAN_RECOVER 69
CAN_REMOTE_REQUEST 92
CAN_REMOTE_RESPONSE 94
CAN_RX 81
CAN_RX_ENH 87
CAN_RX_ENH_FIFO 89
CAN_RX_RANGE 83
CAN_RX_RANGE_FIFO 85
CAN_SETDOWNLOADID 72
CAN_STATUS 70
CAN_TX 74
CAN_TX_ENH 76
CAN_TX_ENH_CYCLIC 78
CAN-Bausteine auf Schicht 2 (RAW-CAN) 64
CAN-Bausteine nach CANopen 130
CAN-Bausteine nach SAE J1939 96
CAN-Buspegel 54
CAN-Datenaustausch 57
CAN-Deklaration (z.B. CR1080) 20

© ifm electronic gmbh | www.ifm.com

CAN-Fehler und Fehlerbehandlung 199
CAN-Funktionen im IEC-User-Zyklus....60
CAN-ID....58
CANopen Begriffe und Implementation....132
CANopen Error-Code 276
CANOPEN_ENABLE....165
CANOPEN_GETBUFFERFLAGS 189
CANOPEN_GETEMCYMESSAGES....209
CANOPEN_GETERRORREGISTER....207
CANOPEN_GETGUARDHBERRLIST 191
CANOPEN_GETGUARDHBSTATSLV 193
CANOPEN_GETNMTSTATESLAVE 178
CANOPEN_GETODCHANGEDFLAG 195
CANOPEN_GETSTATE....182
CANOPEN_GETSYNCSTATE....187
CANOPEN_NMTSERVICES....176
CANOPEN_READOBJECTDICT 196
CANOPEN_SDOREAD 167
CANOPEN_SDOREADMULTI 169
CANOPEN_SDOWRITE 171
CANOPEN_SDOWRITEMULTI....173
CANOPEN_SENDEMCYMESSAGE 205
CANOPEN_SETSTATE 179
CANOPEN_SETSYNCSTATE 185
CANOPEN_WRITEOBJECTDICT 197
CANopen-Bausteine zur Fehlerbehandlung 204
CANopen-Master....134
Register [CAN-Parameter]....136
CANopen-Netzwerk-Management 175
CANopen-Netzwerkvariablen 158
CANopen-Netzwerkvariablen konfigurieren 158
CANopen-Slave....151
Register [CAN Parameter]....138
CANopen-Slave konfigurieren....152
CANopen-Slaves einfügen und konfigurieren....137
CANopen-Status des Knotens....148, 275
CANopen-Tabellen....265
CANopen-Unterstützung durch CoDeSys....132
CAN-Schnittstellen....50
CAN-Stack....302
CCF....302
CiA....302
CiA DS 304....302
CiA DS 401....302
CiA DS 402....302
CiA DS 403....302
CiA DS 404....302
CiA DS 405....302
CiA DS 406....302
CiA DS 407....303
COB-ID....303
CoDeSys....303
CoDeSys-CANopen-Bibliotheken....298
CPU-Frequenzen....41
CRC....303
CSV-Datei....303
CURRENT_CONTROL....233

D

Das Objektverzeichnis des CANopen-Masters....150
Dateien für Betriebssystem / Laufzeitsystem....297
Dateien und Bibliotheken im Gerät installieren....294
Dateien verwalten....248
Dateifunktionen....248
Daten im Speicher sichern, lesen und wandeln....237
Daten verwalten....235
Datentyp....303
Datenzugriff und Datenprüfung....244
DC....303
Der Master zur Laufzeit....142
Diagnose....304
Diagnose der Ausgänge konfigurieren....30
Diagnose der Eingänge aktivieren....23
Diagnose-Deckungsgrad....304
Digital- und PWM-Ausgänge....29
Digitaleingänge....24
Dither....304
diversitär....304
DRAM....304
DTC....305

E

ECU....305
EDS-Datei....305
Ein CANopen-Projekt erstellen....135
Ein-/Ausgangs-Funktionen....211
Eingänge konfigurieren....23
Eingangsgruppe I0 (%IX0.0...%IX0.3)....27
Eingangsgruppe I1 (%IX0.4...%IX0.7)....28
Eingangsgruppe I2 (%IX0.8...%IX0.11)....28
Eingangswerte verarbeiten....211
Einstellen der Knotennummer und der Baud-Rate eines CANopen-Slave....157
Einstellung in CR040n....254
Einstellung in CR0451....255
Einstellungen in den globalen Variablenlisten....160
Einstellungen in den Zielsystemeinstellungen....159
Embedded Software....305
EMCY....305
EMCY-Fehlercode....200
Emergency-Messages durch das Applikations-Programm senden....157
Emergency-Nachrichten....276
Empfangen von Emergency-Messages....143
EMV....305

© ifm electronic gmbh | www.ifm.com

ERROR-Zustand....37
Erstfehler-Eintrittszeit....305
Ethernet....305
EUC....305
Extended CAN-IDs....131

F

FASTCOUNT....216
FATAL-ERROR-Zustand....37
FB, FUN, PRG in CoDeSys....44
FB-Rückmeldungen (Übersicht)....65
FBs für RAW-CAN (Nachrichten empfangen)....80
FBs für RAW-CAN (Nachrichten senden)....73
FBs für RAW-CAN (Remote-Zugriff)....91
FBs für RAW-CAN (Verwaltung)....66
Fehlanwendung....305
Fehler....306
Fehlercodes und Diagnoseinformationen....48
Fehler-Toleranzzeit....306
FiFo....306
FiFo-Begrenzungen....64
Firmware....306
FLASH_INFO....239
FLASH_READ....240
Flash-Speicher....306
FMEA....306
FRAM....307
Funktionale Sicherheit....307
Funktionalität der CANopen-Slave-Bibliothek....151
Funktions-Code / Predefined Connectionset....268
Funktionskonfiguration der Ein- und Ausgänge....22

G

Gebrauchsdauer Tm....307
Gefährdung....307
Gerätefehler signalisieren....200
GET_APP_INFO....249
GET_HW_INFO....250
GET_IDENTITY....245
GET_SW_INFO....252
Globale Variablenliste
  Bestätigter Transfer....162
  Ereignisgesteuerte Übertragung....162
  Lesen....162
  Netzwerktyp....161
  Prüfsumme übertragen....162
  Schreiben....162
  Übertragung bei Änderung....162
  Variablen packen....161
  Variablenlistenkennung (COB-ID)....161
  Zyklische Übertragung....162
Grundeinstellungen
  EDS-Datei generieren....152
  Name der Updatetask....152
  Name des Busses....152

H

Heartbeat....307
Heartbeat vom Master an die Slaves....143
Herstellerspezifische Informationen....203
Hinweise....58, 64
Hinweise zur Anschlussbelegung....33
Historie der Anleitung (BC)....7
HMI....307
Hochlauf der CANopen-Slaves....147
Hochlauf des CANopen-Masters....145
Hochlauf des Netzwerks ohne [Automatisch starten]....149

I

ID - Identifier....307
Identifier....199
Identifier nach SAE J1939....98
IDs (Adressen) in CANopen....133, 265
IEC-User-Zyklus....307
ifm weltweit • ifm worldwide • ifm à l'échelle internationale....325
ifm-CANopen-Hilfsbibliotheken Master/Slave....298
ifm-CANopen-SDO-Bausteine....164
ifm-Gerätebibliotheken....297
ifm-Maintenance-Tool nutzen....47
INC_ENCODER....218
INIT-Zustand (Reset)....35
INPUT....212
IP-Adresse....307
ISO 11898....308
ISO 11992....308
ISO 16845....308

J

J1939_DM1RX....128
J1939_DM1TX....119
J1939_ENABLE....103
J1939_GETDABYNAME....107
J1939_NAME....105
J1939_RX....122
J1939_RX_FIFO....124
J1939_RX_MULTI....126
J1939_STATUS....109
J1939_TX....112
J1939_TX_ENH....113
J1939_TX_ENH_CYCLIC....117
J1939_TX_ENH_MULTI....115

© ifm electronic gmbh | www.ifm.com

K

Kategorie (CAT) 308
Klemme 15 308
Konfiguration aller fehlerfrei detektierten Geräte 143
Konfigurationen 16

L

Lebensdauer, mittlere 308
LED 308
LED ansteuern 256
LED, Buzzer, Visualisierung 256
Leistungsgrenzen der Geräte (CANopen) 131
Leistungsgrenzen der Geräte (SAE J1939) 101
Leistungsgrenzen des Geräts 41
Leitungsquerschnitte 56
Link 308
LSB 308

M

MAC-ID 308
Man unterscheidet folgende Fehler: 199
Manuelle Datensicherung 238
Master 308
Master / Slave 131
MEMCPY 242
MMI 309
Mögliche Betriebsarten Ein-/Ausgänge (CR0401) 262
Mögliche Betriebsarten Ein-/Ausgänge (CR0403) 263
MRAM 309
MSB 309
MTBF 309
MTTF 309
MTTFd 309
Muting 309

N

Netzaufbau 53
Netzwerk starten 144
Netzwerk-Management (NMT) 272
Netzwerk-Management-Kommandos 272
Netzwerkzustände 145
NMT 309
NMT-Status 273
NMT-Status für CANopen-Master 146, 273
NMT-Status für CANopen-Slave 147, 274
Node 309
Node Guarding 309
Nodeguarding mit Lifetime-Überwachung 143

O

Obj / Objekt 309
Objekt 0x1001 (Error-Register) 202, 278
Objekt 0x1003 (Error Field) 200
Objektverzeichnis 310
OBV 310
operational 310
OUTPUT 224

P

PC-Karte 310
PCMCIA-Karte 310
PDM 310
PDO 310
PDO-Mapping
  Eigenschaften 140
  Einfügen 140
PDU 310
Performance-Level 310
PERIOD 221
PES 310
PGN 310
Physikalische Anbindung des CAN 53
PID-Regler 311
Piktogramm 311
PL 311
PLCPRG_DELAY 254
PLr 311
Pre-Op 311
prepared 311
Programmierhinweise für CoDeSys-Projekte 44
Programmiersprache, sicherheitsrelevant 311
Programmiersystem einrichten 16
Programmiersystem manuell einrichten 16
Programmiersystem über ifm-Templates einrichten 21
Prozessabbild 312
PWM 312
PWM – Einführung 228
PWM1000 230
PWM-Frequenz 229
PWM-Funktionen 227
PWM-Funktionen und deren Parameter 229
PWM-Kanäle 0...7/11 229
PWM-Signalverarbeitung 228

R

Ratio 312
RAW-CAN 312
Reaktion auf System-Fehler 48
redundant 312
Register [Default PDO-Mapping] 154

©ifm electronic gmbh | www.ifm.com

Register [Grundeinstellungen] 152
Register [PDO-Mapping empfangen] und [PDO-Mapping senden] 140
Register [Service Data Objects] 141
remanent 313
Reset aller konfigurierten Slaves am Bus beim Systemstart 142
Restrisiko 313
Risiko 313
Risikoanalyse 313
Risikobeurteilung 313
Risikobewertung 313
ro 313
RTC 313
Rücksetzen aller Knoten des Netzwerks mit CAN-Telegramm 149
Rückstellung, manuell 314
RUN-Zustand 36
rw 314

S

SAE J1939 314
Schaden 314
Schnelle Eingänge 24
Schutzmaßnahme 314
SCT 314
SD-Card 314
SDO 314
SDO-Abbruch-Code 270
SDO-Kommando-Bytes 269
Selbsttest 315
SET_IDENTITY 246
SET_LED 257
SET_PASSWORD 247
Sicherheitsfunktion 315
Sicherheitshinweise 8, 23
Sicherheits-Normentypen 315
SIL 315
Slave 315
Speichern 45
spezielle ifm-Bibliotheken 299
SPS-Zyklus optimieren 253
SRDO 315
SRP/CS 315
SRVT 316
Start aller fehlerfrei konfigurierten Slaves 143
Starten aller Knoten des Netzwerks mit CAN-Telegramm 149
Starten des Netzwerks mit GLOBALSTART 149
Status-LED 38
Steuerungskonfiguration 15, 316
Steuerungskonfiguration aktivieren (z.B. CR1080) 18
Steuerungskonfigurations-Datei 297
STOP-Zustand 36
Strommessung bei PWM-Kanälen 232
Stromregelung mit PWM 232
Symbole 316
SYNC-Funktionalität 131
Systembeschreibung 10
System-Konfiguration 52
Systemmerker 264
Systemvariable 316
Systemzeit lesen / schreiben 235

T

Target 316
Target einrichten 17
Target-Datei 297
TCP 316
Technisches zu CANopen 130
Template 316
Testrate rt 316
TIMER_READ_US 236
Topologie 49

U

Über diese Anleitung 5
Überblick 131
Überlastschutz 232
Übersicht 48
Übersicht CANopen EMCY-Codes (CR040n) 203
Übersicht CANopen Error-Codes 201, 277
Übersicht der verwendeten Dateien und Bibliotheken 294
Überwachung 316
UDP 316
Unterschiede in der Gerätefamilie 10

V

Verändern des Standard-Mappings durch Master-Konfiguration 156
Verfügbare CAN-Schnittstellen und CAN-Protokolle 50
Verfügbarer Speicher 43
Verfügbarkeit von PWM 227
Verhalten des Watchdog 42
Verteilen der Applikations-Software 47
Verwendung, bestimmungsgemäß 317

W

Was bedeuten die Symbole und Formatierungen? 5
Watchdog 317
Welche Vorkenntnisse sind notwendig? 9
Wichtig! 8
Widerstandsmessung 26
Wie ist diese Dokumentation aufgebaut? 6
wo 317
Wozu dienen die einzelnen Dateien und Bibliotheken? 297

© ifm electronic gmbh | www.ifm.com

Z

Zählerfunktionen zur Frequenz- und Periodendauermessung 215
Zubehör.....13
Zugriff auf den CANopen-Slave zur Laufzeit.....157
Zugriff auf den Status des CANopen-Masters 150
Zugriff auf die OD-Einträge vom Applikations-Programm.....157
Zusammenfassung CAN / CANopen.....59
Zustand, sicher 317
Zustandsdiagramm eines CANopen-Knotens.....175
Zyklisches Senden der SYNC-Message 143
Zykluszeit.....317
Zykluszeit beachten!.....45
Zykluszeit steuern.....253

© ifm electronic gmbh | www.ifm.com

16 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Stand: 2010-10-08 ifm electronic – close to you! 8310

http://www.ifm.com • E-Mail: info@ifm.com

Service-Hotline: 0800 16 16 16 4 (nur Deutschland, Mo...Fr, 07.00...18.00 Uhr)

ifm Niederlassungen • Sales offices • Agences

D	ifm electronic gmbh Vertrieb Deutschland Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0 Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0 Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0 Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0 ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen
A	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector pty ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CN	ifm electronic Co. Ltd. • 201210 Shanghai • Tel. +86 21 / 5027 8559
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel . +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E.P.E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Györ • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., ltd. • Togane-shi, Chiba 283-0826 • Tel. +81 475-50-3003
MAL	ifm electronic Pte. Ltd • 80250 Johor Bahru Johor • Tel. +60 7 / 331 5022
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
P	ifm electronic s.a. • 4430-208 Vila Nova de Gaia • Tel. +351 223 / 71 71 08
PL	ifm electronic Sp. z o.o. • 40-524 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RP	Gram Industrial, Inc. • 1770 Mantilupa City • Tel. +63 2 / 850 22 18
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 512 60 Överlida • Tel. +46 325 / 661 500
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	Sang Chai Meter Co., Ltd. • Bangkok 10 400 • Tel. +66 2 / 616 80 51
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49

Technische Änderungen behalten wir uns ohne vorherige Ankündigung vor.
We reserve the right to make technical alterations without prior notice.
Nous nous réservons le droit de modifier les données techniques sans préavis.

© ifm electronic gmbh | www.ifm.com